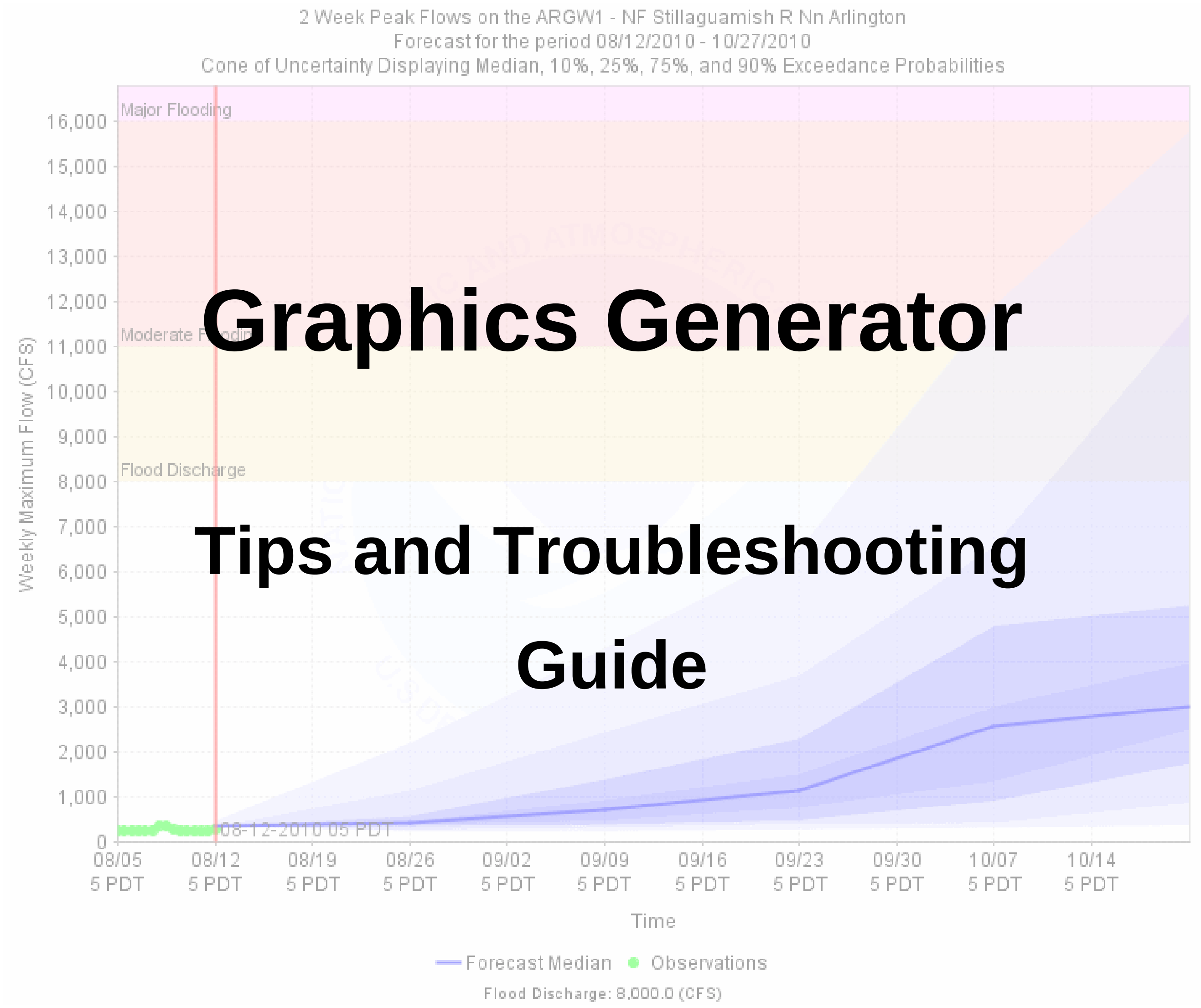
Graphics Generator

Tips and Troubleshooting Guide



Version: OHD-CORE-CHPS-4.4.a

Release Date: 16 May 2016

Table of Contents

1 Overview

This document provides general tips and troubleshooting related to the installation, configuration, and usage of Graphics Generator. The *Graphics Generator Getting Started Manual* should be read prior to reading this document. It is recommended that users read the Table of Contents in order to identify what topics are covered, and read Sections 2.1 and 2.2 prior to designing and building their own products via the **GraphGen Editor Panel**. The other sections of this document should be referred to as needed.

1.1 *Notation*

Within this document, the following notation is used:

- All graphical interface components are **Capitalized and in Bold**.
- All XML snippets are in this font.
- All command line entries are in this font.
- All important terms are *italicized* when first mentioned.

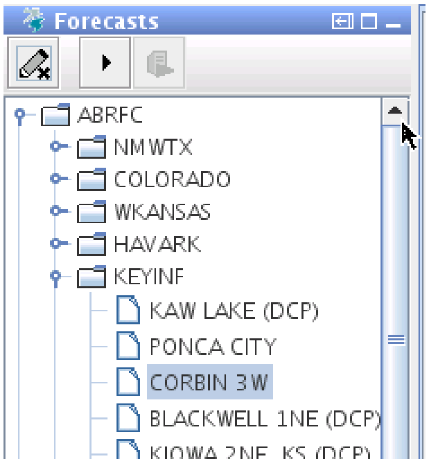
1.2 *Directories of Note*

The following directories will be referred to in the instructions provided below:

- <*region_dir*>: The stand-alone region home directory, typically "##rfc_sa".
- <*configuration_dir*>: The stand-alone Config directory, typically <*region_dir*>/Config.

1.3 *Terminology*

- *active forecast segment*: The current active segment, as selected in the **Forecasts Panel** of the CHPS interface and identified by the segment id set in the configuration file <*configuration file*>/RegionConfigFiles/Topology.xml. For example, for ABRFC, the active forecast segment in this case is CBNK1 (Corbin 3W):



- *appearance modifiers*: See the *Graphics Generator Getting Started Manual*.
- *chart series*: See the *Graphics Generator Getting Started Manual*.
- *identity mapping*: An id-mapping that maps one locationId onto itself; for example <location external="WALN6DEL" internal="WALN6DEL"/>. Such mappings may be necessary for Graphics Generator to acquire needed time series; see the *Graphics Generator Installation Guide* for more information.
- *installation stand-alone*: The stand-alone in which Graphics Generator will be installed.
- *query (FEWS PI-service)*: Instructions for acquiring time series via the FEWS PI-serivce, defined within a PI-service configuration file (under <*configuration_dir*>/PiServiceConfigFiles) by a single timeSeries element; for example:

```
<timeSeries>
    <id>STG</id>
    <timeSeriesSet>
        <moduleInstanceId>ImportIHFSDB</moduleInstanceId>
        <valueType>scalar</valueType>
        <parameterId>STG</parameterId>
        <locationSetId>ImportIHFSDB</locationSetId>
        <timeSeriesType>external historical</timeSeriesType>
        <timeStep unit="hour" multiplier="6" />
        <readWriteMode>read only</readWriteMode>
    </timeSeriesSet
</timeSeries>
```

- *query id*: Identifies a *query* and corresponds to the subelement id within the timeSeries XML element of query. In the example above, the query id is "STG".
- *template*: See the *Graphics Generator Getting Started Manual*.
- *thresholdId*: Shorthand for the levelThresholdId XML element defined for a threshold value in <*configuration_dir*>/RegionConfigFiles/ThresholdValueSets.xml.

2 General Tips

The following tips are generally applicable to all uses of Graphics Generator. Subsections may be referred to in other documents as needed.

2.1 *General Tips for PI-Service XML Configuration*

The following are general tips recommended when constructing FEWS PI-service queries for use in Graphics Generator when building products:

1. *Do* *__not__* *add new queries to the delivered file <configuration_dir>/PiServiceConfigFiles/GraphGen.xml*. Define the queries appropriately based on instructions in the *Graphics Generator Installation Guide*, but do not add new queries. This file may be modified and redelivered by OHD, and it would lessen the amount of work required to merge changes in the existing file with the newly delivered file if no additional queries are added.
2. *Do not reuse query ids in PI-service configuration files*. Confusion will be avoided if each query id is used only one time over all PI-service configuration files.
3. *Every PI-service query should return only a single time series or ensemble per* locationId. This will simplify the instructions identifying which time series or ensemble acquired via the FEWS PI-service is used in the product.

The list above will be expanded in the future as more tips are identified.

2.1.1 Defining Queries That Include Multiple timeSeriesSet XML Elements

In some cases, it may be necessary to specify multiple timeSeriesSet XML elements within a single query in a PI-service configuration file. This is most often the case if the time series returned by a query can have different parameter ids, qualifiers, or time steps for different locations. For example, the following is the default “QINE ESP” query used for the AHPS products, which uses ABRFC as an example:

```
<timeSeries>
  <id>QINE ESP</id>
  <timeSeriesSet>
    <moduleInstanceSetId>ALL_QINE_Forecast</moduleInstanceSetId>
    <valueType>scalar</valueType>
    <parameterId>QINE</parameterId>
    <locationSetId>ImportIHFSDB</locationSetId>
    <timeSeriesType>simulated forecasting</timeSeriesType>
    <timeStep unit="hour" multiplier="6" timeZone="GMT-0" />
    <readWriteMode>read only</readWriteMode>
    <ensembleId>ESP</ensembleId>
    <ensembleMemberIndexRange start="1951" />
  </timeSeriesSet>
</timeSeries>
```

Suppose ABRFC includes some segments for which the streamflow ensemble has a 1-hour time step. Then, the “QINE ESP” query must contain two timeSeriesSet XML elements: one for 6-hour ensembles and one for 1-hour ensembles. For example, consider the following change (duplicating the existing timeSeriesSet and changing the time step, which is highlighted):

```
<timeSeries>
  <id>QINE ESP</id>
  <timeSeriesSet>
    <moduleInstanceSetId>ALL_QINE_Forecast</moduleInstanceSetId>
    <valueType>scalar</valueType>
    <parameterId>QINE</parameterId>
    <locationSetId>ImportIHFSDB</locationSetId>
    <timeSeriesType>simulated forecasting</timeSeriesType>
    <timeStep unit="hour" multiplier="6" timeZone="GMT-0" />
    <readWriteMode>read only</readWriteMode>
    <ensembleId>ESP</ensembleId>
    <ensembleMemberIndexRange start="1951" />
  </timeSeriesSet>
  <timeSeriesSet>
    <moduleInstanceSetId>ALL_QINE_Forecast</moduleInstanceSetId>
    <valueType>scalar</valueType>
    <parameterId>QINE</parameterId>
    <locationSetId>ImportIHFSDB</locationSetId>
    <timeSeriesType>simulated forecasting</timeSeriesType>
    <timeStep unit="hour" multiplier="1" timeZone="GMT-0" />
    <readWriteMode>read only</readWriteMode>
    <ensembleId>ESP</ensembleId>
    <ensembleMemberIndexRange start="1951" />
  </timeSeriesSet>
</timeSeries>
```

The problem with the change above is that the same moduleInstanceSet and locationSet is used in both timeSeriesSet XML elements. For a given locationId, then, the FEWS PI-service may return an ensemble of 6-hourly time series and an ensemble of 1-hourly time series. One of the two ensembles will contain non-missing time series while the other ensemble will contain only missing time series (since the ensemble is either 6-hours or 1-hours, not both). This violates item 3 in Section 2.1, above: *Every PI-service query should return only a single time series or ensemble per* locationId.

Use the **Graphics Generator Editor Panel** as described in Section 2.5, clicking on **Show table** as described in Step 5 to view the raw time series provided by the FEWS PI-service. Make sure it is not returning too many time series for each location.

When adding multiple timeSeriesSet XML elements, therefore, pay careful attention to ensure that only one time series or ensemble is returned for each locationId. In the example above, the simplest solution is to break down the location set “ImportIHFSDB” into two location sets: “ESP_Locations_1-hour” and “ESP_Locations_6-hour”. Define those two location sets appropriated and then define the query as follows (the changes are highlighted):

```
  <timeSeries>
    <id>QINE ESP</id>
    <timeSeriesSet>
      <moduleInstanceSetId>ALL_QINE_Forecast</moduleInstanceSetId>
      <valueType>scalar</valueType>
      <parameterId>QINE</parameterId>
      <locationSetId>ESP_Locations_6-hour</locationSetId>
      <timeSeriesType>simulated forecasting</timeSeriesType>
      <timeStep unit="hour" multiplier="6" timeZone="GMT-0" />
      <readWriteMode>read only</readWriteMode>
      <ensembleId>ESP</ensembleId>
      <ensembleMemberIndexRange start="1951" />
    </timeSeriesSet>
    <timeSeriesSet>
      <moduleInstanceSetId>ALL_QINE_Forecast</moduleInstanceSetId>
      <valueType>scalar</valueType>
      <parameterId>QINE</parameterId>
      <locationSetId>ESP_Locations_1-hour</locationSetId>
      <timeSeriesType>simulated forecasting</timeSeriesType>
      <timeStep unit="hour" multiplier="1" timeZone="GMT-0" />
      <readWriteMode>read only</readWriteMode>
      <ensembleId>ESP</ensembleId>
      <ensembleMemberIndexRange start="1951" />
    </timeSeriesSet>
  </timeSeries>
```

With this change, each location is only included in one of the two timeSeriesSet XML elements, so that only one ensemble is returned for each.

2.1.2 Making Use of ModuleInstanceSets and LocationSets

To minimize the amount of configuration required to specify FEWS PI-service queries, you are encouraged to define module instance sets and location sets as needed. They are defined in the files (respectively):

…/Config/RegionConfigFiles/ModuleInstanceSets.xml
…/Config/RegionConfigFiles/LocationSets.xml

This is particularly true if the modules instances that generate time series to be returned in the query execute various models or the locationIds tend to vary between data type or module instance.

Consider the following PI-service query which is used to acquire observed time series that are input to an ensemble post-processing model (MARFC time series are used in this example):

```
<timeSeries>
  <id>OBSERVED FLOW</id>
  <timeSeriesSet>
    <moduleInstanceId>ADJUSTQ_WALN6DEL_WALN6DEL_Forecast</moduleInstanceId>
    <valueType>scalar</valueType>
    <parameterId>QINE</parameterId>
    <locationId>WALN6DEL</locationId>
    <timeSeriesType>simulated forecasting</timeSeriesType>
    <timeStep unit="hour" multiplier="6"/>
    <readWriteMode>read only</readWriteMode>
    <ensembleId>main</ensembleId>
  </timeSeriesSet>
  <timeSeriesSet>
    <moduleInstanceId>ADDSUB_CNNN6DEL_ADDSPILL_Forecast</moduleInstanceId>
    <valueType>scalar</valueType>
    <parameterId>QIN</parameterId>
    <locationId>CNNN6TOT</locationId>
    <timeSeriesType>simulated forecasting</timeSeriesType>
    <timeStep unit="hour" multiplier="6"/>
    <relativeViewPeriod unit="hour" start="-120" startOverrulable="true" end="0"/>
    <readWriteMode>read only</readWriteMode>
  </timeSeriesSet>
</timeSeries>
```

In this case, the first time series applies to segment WALN6DEL and uses output from an ADJUSTQ operation as a source of observed streamflow. The second time series, on the other hand, provides an observed total inflow to reservoir within the CNNN6DEL segment. The module instance is an ADDSUB operation and the locationId is CNNN6TOT.

This example is only covering two of the segments within MARFC. In general, the time series that must be returned by the query could be output from any of several types of models and could have parameterIds QINE, SQIN, QIN, or QME. The locationIds will also vary from segment to segment, in some cases matching the id of the segment and in other cases not.

To simplify the configuration, it is recommended that the following module instance set be created:

```
<moduleInstanceSet id="EnsPost_Observed_Input" name="EnsPost observed input">
  <moduleInstanceId>ADJUSTQ_WALN6DEL_WALN6DEL_Forecast</moduleInstanceId>
  <moduleInstanceId>ADDSUB_CNNN6DEL_ADDSPILL_Forecast</moduleInstanceId>
</moduleInstanceSet>
```

It is also recommended that the following location set be added:

```
<locationSet id="Catchments_HEFS_UpperDelaware_Export">
  <locationId>WALN6DEL</locationId>
  <locationId>CNNN6TOT</locationId>
</locationSet>
```

Then, define the query as follows:

```
<timeSeries>
  <id>OBSERVED FLOW</id>
  <timeSeriesSet>
    <moduleInstanceSetId>EnsPost_Observed_Input</moduleInstanceSetId>
    <valueType>scalar</valueType>
    <parameterId>QINE</parameterId>
    <locationSetId>Catchments_HEFS_UpperDelaware_Export</locationSetId>
    <timeSeriesType>simulated forecasting</timeSeriesType>
```

```
        <timeStep unit="hour" multiplier="6"/>
        <readWriteMode>read only</readWriteMode>
        <ensembleId>main</ensembleId>
    </timeSeriesSet>
    <timeSeriesSet>
        <moduleInstanceSetId>EnsPost_Observed_Input</moduleInstanceSetId>
        <valueType>scalar</valueType>
        <parameterId>QIN</parameterId>
        <locationSetId>Catchments_HEFS_UpperDelaware_Export</locationSetId>
        <timeSeriesType>simulated forecasting</timeSeriesType>
        <timeStep unit="hour" multiplier="6"/>
        <relativeViewPeriod unit="hour" start="-120" startOverrulable="true" end="0"/>
        <readWriteMode>read only</readWriteMode>
    </timeSeriesSet>
</timeSeries>
```

Then, when Graphics Generator requests a time series from the FEWS PI-service for this query for location with id WALN6DEL, the FEWS PI-service will recognize that the only time series for location WALN6DEL from the query above is the one with parameterId QINE output by the module ADJUSTQ_WALN6DEL_WALN6DEL_Forecast. It will return that one time series.

For the example above, it was also necessary to define an id-mapping, since the locationId CNNN6TOT is not identical to the segment id, CNNN6DEL, and other queries needed to generate products for segment CNNN6DEL returned time series with other locationIds, such as CNNN6DEL and CNNN6TIF. Product installation instructions will include details about when it is necessary to define id-mappings.

2.2 *General Tips for Product Design*

The following are general tips recommended when designing and building a product via the **GraphGen Editor Panel**:

1. *If a chart series is to appear in more than one product, it should be defined in a referenced template*. For example, if observed stream flow and river stage data will be shown as green circles in more than one of the products produced via Graphics Generator, then a referenced template should be defined to draw that series. That referenced template can then be added in product templates as needed in order to display observed streamflow or river stage data.

2. *Do not add appearance modifiers to referenced templates*. Referenced template appearance modifiers are combined in order to create a base-default set of appearance modifiers for a product. This can be confusing if many referenced templates are used in a single product. For simplicity, do not include appearance modifiers in the definition of referenced templates. There may be some exceptions to this rule; use your best judgment.

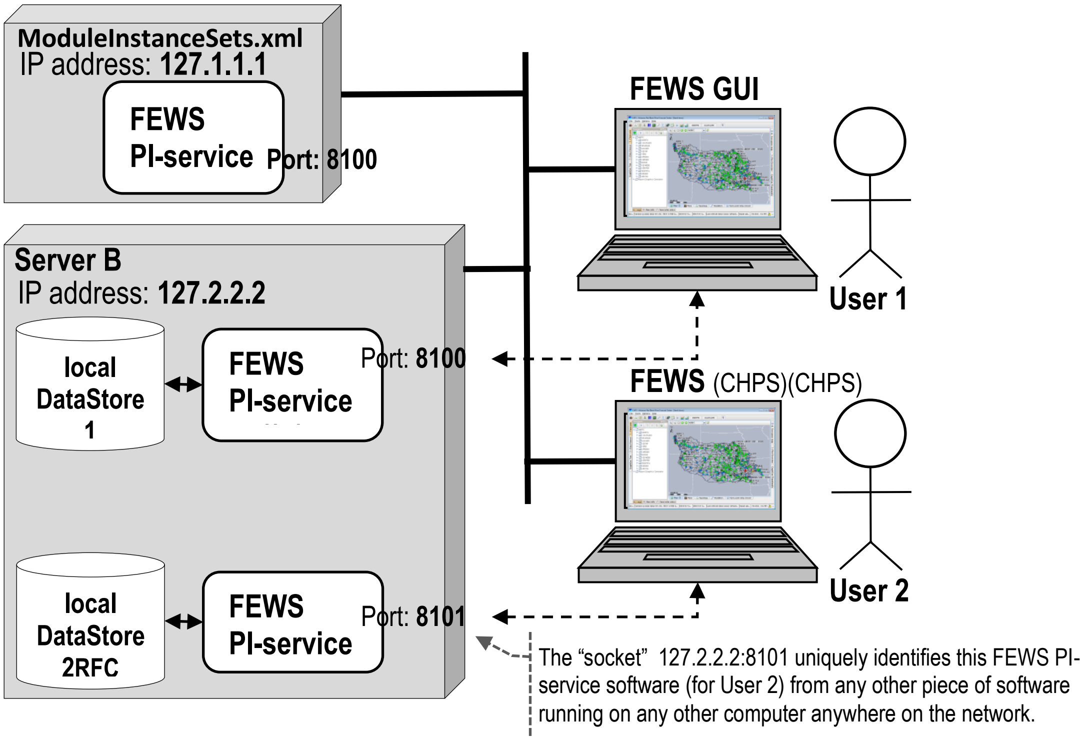
3. *Do not include references to other referenced templates within the definition of a referenced template.* Product templates can be designed to include referenced templates, but, for simplicity, referenced templates should not include other referenced templates. Again, however, there may be some exceptions to this rule; use your best judgment.

The list above will be expanded in the future as more tips are identified.

2.3 *Setting the FEWS PI-Service Port Number for Interactive Use*

Graphics Generator makes use of the FEWS Published Interface (PI)-service. However, before Graphics Generator can make use of the FEWS PI-service, it must be configured correctly. The below describes the problem of identifying the correct connection and how to direct Graphics Generator to use the correct connection.

IP addresses enable computers to be uniquely addressed. Since each computer has its own unique IP address; messages can be correctly delivered (from one computer to the next) as long as the message contains the destination's IP address. However, with **multiple pieces of software on a single computer**, ports are also required:



The "socket" 127.2.2.2:8101 uniquely identifies this FEWS PI-service software (for User 2) from any other piece of software running on any other computer anywhere on the network.

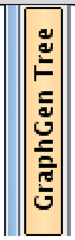
In the above figure, the FEWS interface for User 2 needs to send a message to the FEWS PI-service software on Server B. However, there are two copies of the PI-service running on Server B. Which copy of the PI-service will receive the message? Using only the IP address for Server B (127.2.2.2) will not indicate which of the two PI-services will receive the message. Moreover, we do not want User 2 changing data in the localDataStore that belongs to User 1. Consequently, we need an addressing mechanism that uniquely identifies both computers **and** FEWS instances running on those computers. Port numbers supply the additional piece of information that uniquely identifies a single FEWS instance on a computer. (The combination of an IP address and a port number is often referred to as a "socket".)

When the first user (on a particular computer) starts FEWS, it automatically starts a FEWS PI-service for that user and assigns the PI-service software a port number of 8100 (unless configured otherwise through the global property, portNumber). However, when a second user on the same computer attempts to run FEWS, FEWS recognizes that another user is already “using” port number 8100 and automatically assigns the next user PI-service port number 8101. The third user will get 8102, and so on.

Unfortunately, Graphics Generator has no way of knowing which PI-service port number was assigned to an instance of FEWS. The FEWS software does not currently provide a mechanism for FEWS explorer plug-in to ask what the current user’s FEWS instance port number is. Consequently, **the FEWS PI-service connection must be manually defined by users**.

To ensure a proper connection to the PI-service, the following should be executed immediately after starting the CHPS interface that has Graphics Generator installed:

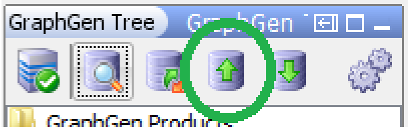
#	**Action**
1	To find your PI-service port number, check the **Logs Panel** for lines similar to the following: 11-04-2010 11:16:08 INFO - OHD FEWS explorer plug-in software establishing connection to CHPS FewsPiServiceImpl on localHost : 8100... 11-04-2010 11:16:01 INFO - Started FewsPiServiceImpl on localHost : 8101 11-04-2010 11:16:01 WARN - Failed to start: SocketListener0@0.0.0.0:8100 In the above example, note the yellow highlighted number 8100. 8100 was the port number that the Graphics Generator attempted to connect to in order to access the PI-service. Note also the above warning: “WARN – Failed to start: SocketListener0@0.0.0.0:8100”. This is an indicator that port 8100 was not available because another user is already using it. Look for the text “Started FewsPiServiceImpl” which indicates the automatically assigned port number, shown highlighted in green above. In this case, 8101 is the port number of the PI-service started for **your FEWS session** and 8101 is the port number Graphics Generator should use. (Note that in this case the new port number was 8101. This may not always be true since it is not always obvious how many FEWS users are using the same computer.) **You should always check to see which port number was assigned to your FEWS PI-service**. If your assigned port number is not the default (8100), as is the case above, the HEFS GUI components will fail to connect to the PI-service or will connect to the *wrong* PI-service. **If your port number is not 8100, then continue to Step 2. Otherwise, if your port number is 8100 there is no need to make a correction and the steps below can be skipped.**

#	Action
2	Click on the  (tab) on the left. Click on the **Reconnect to CHPS PI-service Button** , ✓, to set the port number.
3	In the **Enter Port Number Dialog** that opens, enter the correct port number (green highlighted number in Step 1 above.) and click **OK**. After a brief delay, a connection will be established (✓ button will display) or an error message will be displayed if a problem occurred (❗ button will display).

2.4 *Saving and Uploading Products*

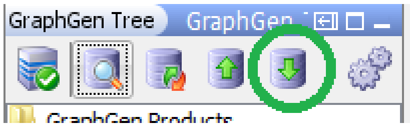
When making changes to products or settings in Graphics Generator, changes are made only in the *local area* products and settings file, being saved in an XML file on the file system. This local area file is intended for local use only. If the changes are intended to be shared or permanent, you should upload the changes to the *central area* file. See the *Graphics Generator Getting Started* manual.

To upload changes to the central area products and settings file, so that the changes can become shared, in the **GraphGen Tree Panel**, click on the **Upload Products Button**:



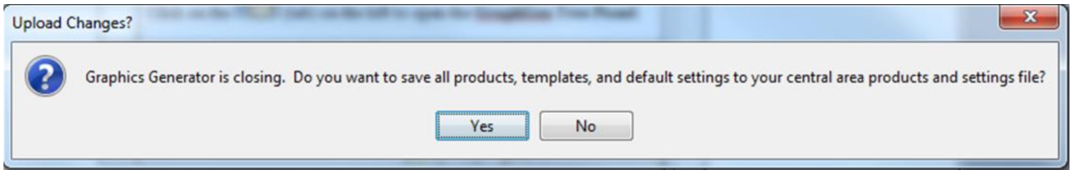
When the confirmation dialog opens, click **Yes**.

You can also download the products and setting from the central area to the local area, discarding products and settings currently defined in the local area. To do so, in the **GraphGen Tree Panel**, click on the **Download Products Button**:



When the confirmation dialog opens, click **Yes**.

You will be notified when CHPS is shut down without uploading changes to the central area products and settings file and given an opportunity to uploaded changes at that time click **Yes** to upload changes to the central area products and settings file:



Upload Changes?

Graphics Generator is closing. Do you want to save all products, templates, and default settings to your central area products and settings file?

Whenever the **Upload Products Button** is clicked, after copying the file OHD_GRAPHGEN_PRODUCTS_AND_SETTINGS.xml from the local area to the central area (overwriting the existing copy), a date-stamped copy of the file is created in the central area:

OHD_GRAPHGEN_PRODUCTS_AND_SETTINGS.xml.yyyyMMdd_HHmmss

As stated in *Graphics Generator Getting Started*, up to 50 such files can be stored in the central area directory. If the count exceeds 50, then the oldest files (by date-stamp) are removed until the count is reduced to 50. This central area history allows for users to recover referenced and product templates as well as settings that have been modified or removed. No tools are provided with Graphics Generator to facilitate this: it is up to the users to determine which templates to recover, identify which history file contains those templates, and recover them by creating files that can be imported by Graphics Generator.

The recommended method to use for searching the history files are basic Linux tools grep and diff, as well as any text editor/viewer. For example, to see a list of all product identifiers in a file, do the following:

```
grep "<productId>" <date-stamped file>
```

The grep command can also be used to determine which date-stamped files include a product with a specific identifier; for example:

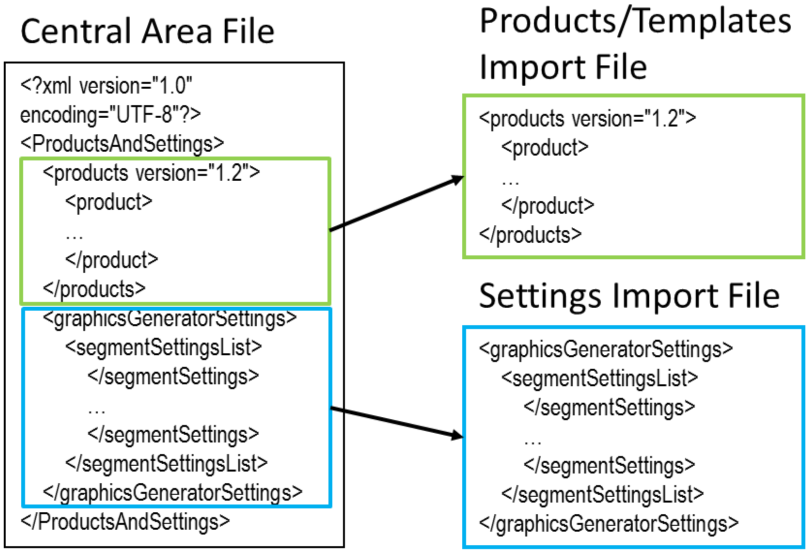
```
grep "<productId>desired identifier</productId>" *.xml.*
```

The diff (or vimdiff) command can be used to determine which changes were made from one file to the next and relative to the primary central area file:

```
diff <date-stamped file 1> <date-stamped file 2>
```

Old modified or removed settings can also be found using grep and diff in an analogous manner or through the use of a text editor. Settings are stored by segment or group in XML elements called segmentSettings with the attribute segmentId used to identify the segment or group.

Once it is identified which date-stamped file includes the referenced and product templates and/or settings that must be recovered, the safest way to recover those templates and settings is to copy the XML elements defining the templates and/or settings into one file containing the referenced and product templates and one file containing the settings. The formats of the XML files are as follows:



Central Area File

```
<?xml version="1.0"
encoding="UTF-8"?>
<ProductsAndSettings>
  <products version="1.2">
    <product>
    …
    </product>
  </products>
  <graphicsGeneratorSettings>
    <segmentSettingsList>
      </segmentSettings>
      …
      </segmentSettings>
    </segmentSettingsList>
  </graphicsGeneratorSettings>
</ProductsAndSettings>
```

Products/Templates Import File

```
<products version="1.2">
  <product>
  …
  </product>
</products>
```

Settings Import File

```
<graphicsGeneratorSettings>
  <segmentSettingsList>
    </segmentSettings>
    …
    </segmentSettings>
  </segmentSettingsList>
</graphicsGeneratorSettings>
```

For referenced and product templates, extract the pertinent product XML elements from the central area file and paste them into a properly formatted products/templates import file, as shown above. Construct the settings import file in a similar manner, copying the segmentSettings XML elements as needed following the format shown above for a settings import file. Section 2.7.2 then provides instructions for importing the referenced and product templates and settings.

Alternatively, a basic copy command,

```
cp <date-stamped file> OHD_GRAPHGEN_PRODUCTS_AND_SETTINGS.xml
```

can be used if it is acceptable to lose all changes made to the referenced and product templates and settings since the date of the date-stamped file. In that case, the user should download the central area to their SA (click on the **Download Button** in the **GraphGen Tree Panel**) afterwards in order to access the recovered referenced and product templates and settings.

- XML version (see the version attribute associated with XML element products in the examples above) is only a concern if a version of Graphics Generator is backed-out. Graphics Generator will always be written to be backward compatible, so that the latest installed version of Graphics Generator can process XML written by any earlier release.

- Within the XML schema used by Graphics Generator, the element defining a referenced or product template is the product XML element. The element is called “product” since Graphics Generator originally used that term as “template” is used now. The XML schema was never updated to account for this change in terminology, since that would have made all existing central and local area files invalid.

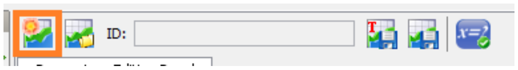
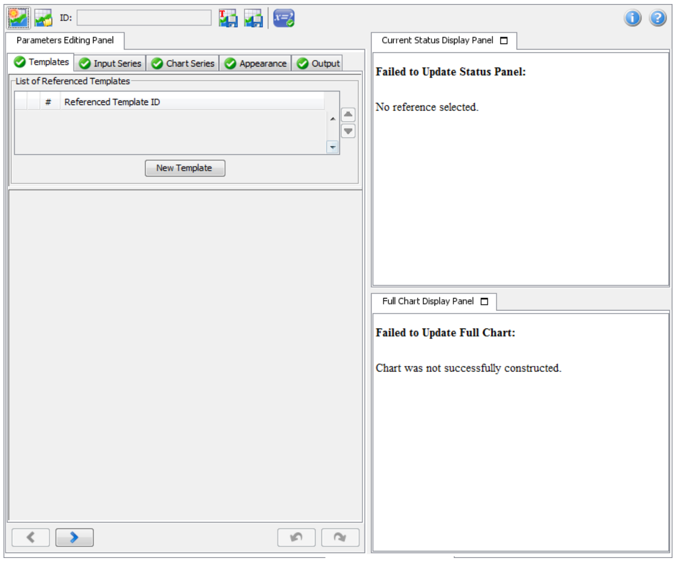
- In general, if the settings that are missing pertain to an entire segment, then copy that segment’s settings from the central area file to the import file following the format above. However, if it is one part of those settings that must be recovered, such as a particular argument that must be overridden, it is recommended that you manually re-enter those settings via the **Modify Settings Dialog** accessible from the **GraphGen Tree Panel**. It is not easy, in that situation, to construct an appropriate settings import file.

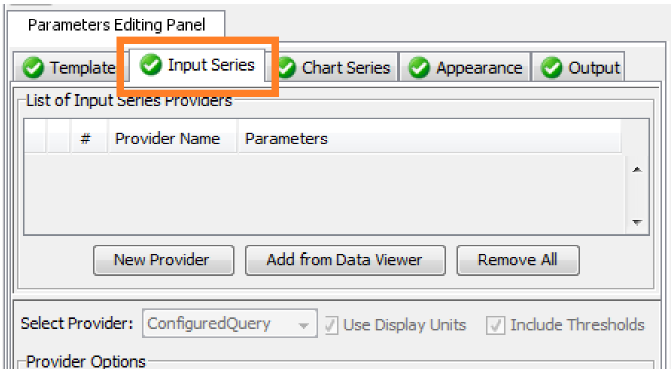
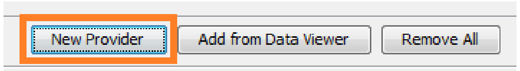
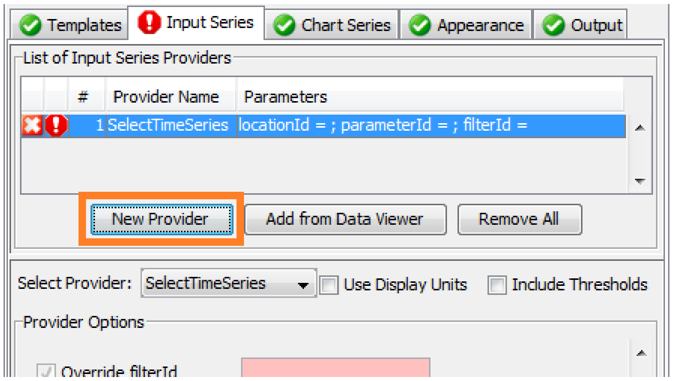
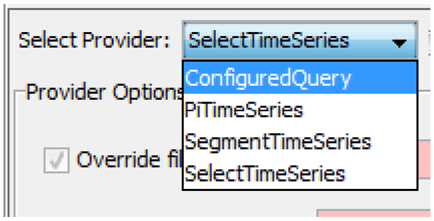
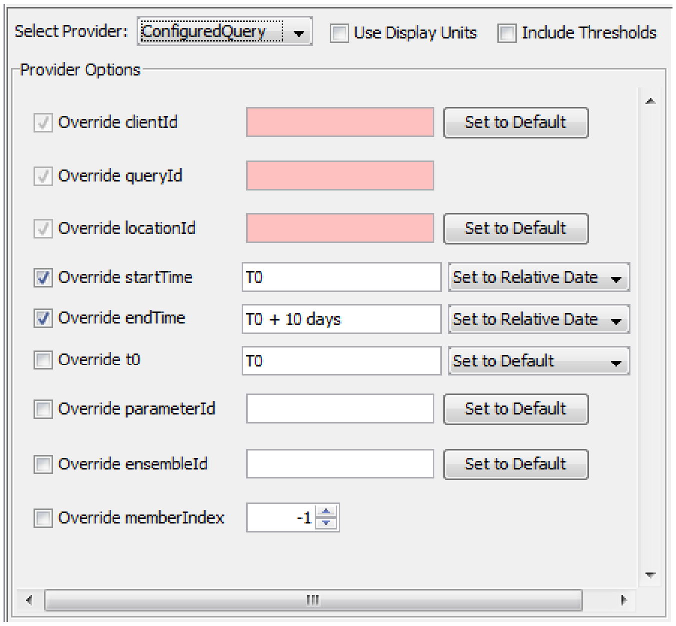
2.5 *Checking the PI-Service Configuration Using Graphics Generator*

The queries defined in configuration of files under <*configuration_dir*>/PiServiceConfigFiles can be checked using the **GraphGen Editor Panel**. The steps are described below. Prior to performing those steps, the user must do the following:

1. Make the desired changes to a FEWS PI-service configuration file, such as GraphGen.xml.
2. Confirm that the localDataStore for the stand-alone that will be used to check the configuration has the desired data and, if the data is created by a forecast workflow, then the workflow was approved. This can be done using the **Database Viewer** in CHPS.
3. Start CHPS.

After starting CHPS, to test the PI-service configuration, do the following and verify the expected results:

#	Action	Expected Results
1	Click on the **GraphGen Editor Button**, , in the CHPS interface main toolbar. After the **GraphGen Editor Panel** opens, click on the **New Button**, , in the toolbar:  	If not already empty, this will create a new, unnamed empty product for editing:  ID: Parameters Editing Panel Templates Input Series Chart Series Appearance Output List of Referenced Templates # Referenced Template ID New Template Current Status Display Panel Full Chart Display Panel **Failed to Update Status Panel:** No reference selected. **Failed to Update Full Chart:** Chart was not successfully constructed.

#	Action	Expected Results
2	Click on the **Input Series Tab**.	The **Input Series Panel** will become active:  Parameters Editing Panel Template Input Series Chart Series Appearance Output List of Input Series Providers # Provider Name Parameters New Provider Add from Data Viewer Remove All Provider Options Select Provider: ConfiguredQuery ☑ Use Display Units ☑ Include Thresholds
3	Click the **New Provider Button**:  	A new input series provider will be added to the temporary product of type "SelectTimeSeries":  Templates Input Series Chart Series Appearance Output List of Input Series Providers # Provider Name Parameters New Provider Add from Data Viewer Remove All Provider Options 1 SelectTimeSeries locationId = ; parameterId = ; filterId = Select Provider: SelectTimeSeries □ Use Display Units □ Include Thresholds ☑ Override filterId
4	From the **Select Provider Choice Box**, select "ConfiguredQuery":  Select Provider: SelectTimeSeries ConfiguredQuery PiTimeSeries SegmentTimeSeries SelectTimeSeries Provider Options Override fi	The provider will change to one of type "ConfiguredQuery":  Select Provider: ConfiguredQuery □ Use Display Units □ Include Thresholds ☑ Override clientId ☑ Override queryId ☑ Override locationId ☑ Override startTime T0 Set to Relative Date ☑ Override endTime T0 + 10 days Set to Relative Date □ Override t0 T0 Set to Default □ Override parameterId □ Override ensembleId □ Override memberIndex -1

<table>
<tr><th>#</th><th>Action</th><th>Expected Results</th></tr>
<tr><td>5</td><td>

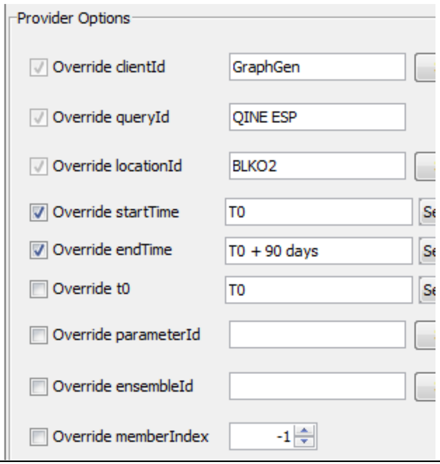
Modify the fields as appropriate for testing the configured PI-service query. The required fields are as follows:

- **clientId**: The name of the file under <*configuration_dir*>/PiServiceConfigFiles without the .xml extension. For example, “GraphGen” corresponds to the GraphGen.xml file.
- **queryId**: The id XML element value of the query to check.
- **locationId**: Restricts the returned time series to those with the specified locationId.
- **startTime**: The first time for which data will be acquired. This can be set relative to the system time, ‘T0’, or can be a fixed date, ‘mm-dd-yyyy hh:mm:ss’.
- **endTime**: The last time for which data will be acquired. See **startTime**.

The optional fields are as follows (for the field to be used, check its checkbox first):

- **t0**: Do not use this field.
- **parameterId**: Restricts the returned time series to those with the specified parameterId.
- **ensembleId**: Restricts the returned time series to those with the specified ensembleId.
- **memberIndex**: Restricts the returned time series to those with the specified ensemble member index.

For example, these settings will execute the ‘QINE ESP’ query within GraphGen.xml for ABRFC’s segment BLKO2:

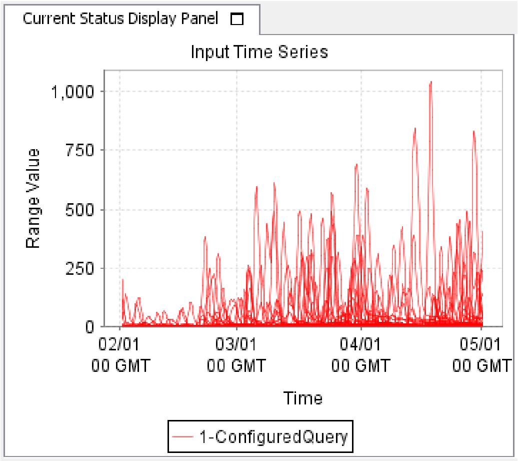


Provider Options

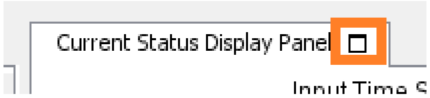
☑ Override clientId: GraphGen
☑ Override queryId: QINE ESP
☑ Override locationId: BLKO2
☑ Override startTime: T0
☑ Override endTime: T0 + 90 days
☐ Override t0: T0
☐ Override parameterId:
☐ Override ensembleId:
☐ Override memberIndex: -1

</td><td>

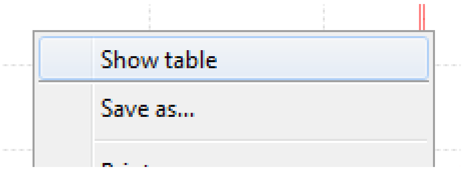
If the FEWS PI-service query is properly specified, then time series will be displayed in the **Current Status Display Panel** (upper right of the **GraphGen Editor Panel**) indicating what was found (example is for the ‘QINE ESP’ query in GraphGen.xml for ABRFC’s segment BLKO2):



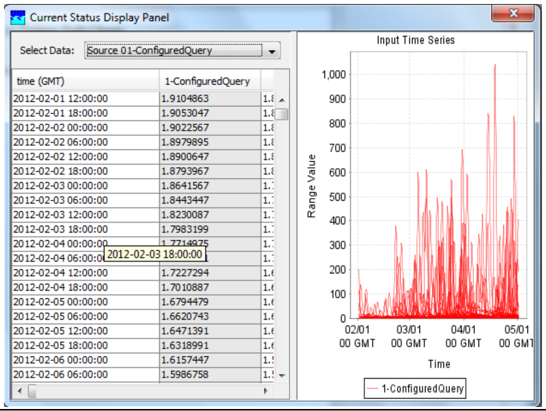
To view the raw numbers, undock the **Current Status Display Panel** by clicking on the □ in the tab



and make the panel larger. Next, right click on the plot, and select **Show table** from the pop-up menu:



The numbers will be displayed on a table to the left of the chart:

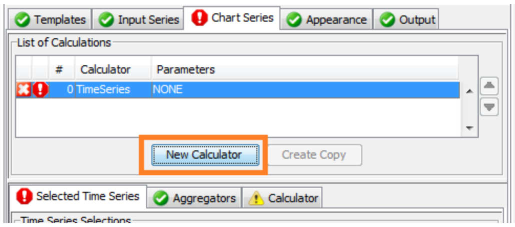


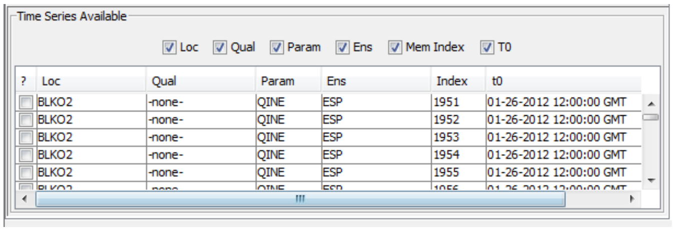
Current Status Display Panel

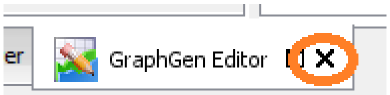
Select Data: Source 01-ConfiguredQuery

time (GMT) — 1-ConfiguredQuery
2012-02-01 12:00:00 — 1.9104863
2012-02-01 18:00:00 — 1.9053047
2012-02-02 00:00:00 — 1.9022567
2012-02-02 06:00:00 — 1.8979895
2012-02-02 12:00:00 — 1.8900647
2012-02-02 18:00:00 — 1.8793967
2012-02-03 00:00:00 — 1.8641567
2012-02-03 06:00:00 — 1.8443447
2012-02-03 12:00:00 — 1.8230087
2012-02-03 18:00:00 — 1.7983199
2012-02-04 00:00:00 — 1.7714975
2012-02-04 06:00:00 — [illegible]
2012-02-04 12:00:00 — 1.7227294
2012-02-04 18:00:00 — 1.7010887
2012-02-05 00:00:00 — 1.6794479
2012-02-05 06:00:00 — 1.6620743
2012-02-05 12:00:00 — 1.6471391
2012-02-05 18:00:00 — 1.6318991
2012-02-06 00:00:00 — 1.6157447
2012-02-06 06:00:00 — 1.5986758

</td></tr>
</table>

<table>
<tr><th>#</th><th>Action</th><th>Expected Results</th></tr>
<tr><td>6</td><td>To view the identifying information about the time series, click on the <b>Chart Series Tab</b> to make the <b>Chart Series Panel</b> active, and click on the <b>New Calculator Button</b>:


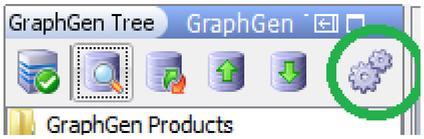
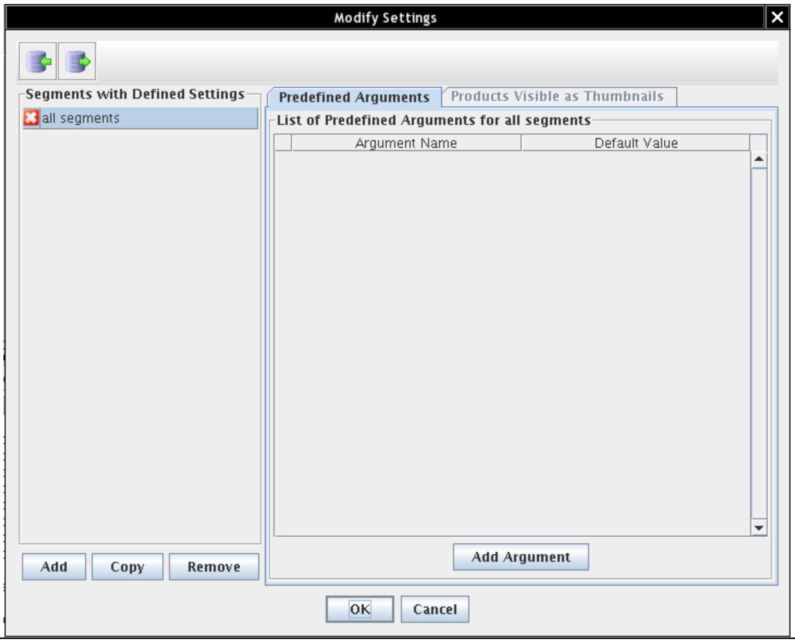
<table><tr><th>#</th><th>Calculator</th><th>Parameters</th></tr><tr><td>0</td><td>TimeSeries</td><td>NONE</td></tr></table></td><td>The <b>Time Series Available Table</b> at the bottom of the <b>Chart Series Panel</b> will display the time series identifying information:


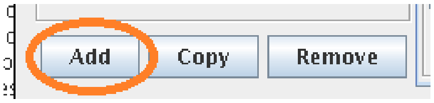
<table><tr><th>?</th><th>Loc</th><th>Qual</th><th>Param</th><th>Ens</th><th>Index</th><th>t0</th></tr><tr><td>□</td><td>BLKO2</td><td>-none-</td><td>QINE</td><td>ESP</td><td>1951</td><td>01-26-2012 12:00:00 GMT</td></tr><tr><td>□</td><td>BLKO2</td><td>-none-</td><td>QINE</td><td>ESP</td><td>1952</td><td>01-26-2012 12:00:00 GMT</td></tr><tr><td>□</td><td>BLKO2</td><td>-none-</td><td>QINE</td><td>ESP</td><td>1953</td><td>01-26-2012 12:00:00 GMT</td></tr><tr><td>□</td><td>BLKO2</td><td>-none-</td><td>QINE</td><td>ESP</td><td>1954</td><td>01-26-2012 12:00:00 GMT</td></tr><tr><td>□</td><td>BLKO2</td><td>-none-</td><td>QINE</td><td>ESP</td><td>1955</td><td>01-26-2012 12:00:00 GMT</td></tr></table></td></tr>
<tr><td>7</td><td>Close the <b>GraphGen Editor Panel</b> by clicking on the X next to its name in the tab:

 </td><td>The <b>GraphGen Editor Panel</b> will close.</td></tr>
</table>

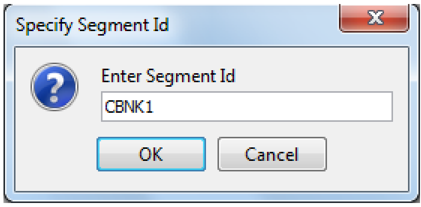
The steps above can be used to verify the time series returned by the FEWS PI-service for use in Graphics Generator. This tests both the content of the file GraphGen.xml file, as well as the id-mappings in the IdExportPiService.xml file, as described in the next section, which modify the time series header XML elements, including locationId, parameterId, ensembleId, and qualifierId, returned by the FEWS PI-service.

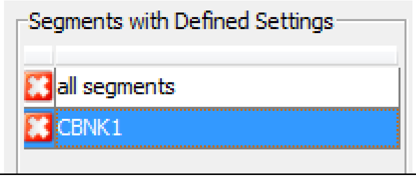
2.6 *Making Products Not Visible as Thumbnails*

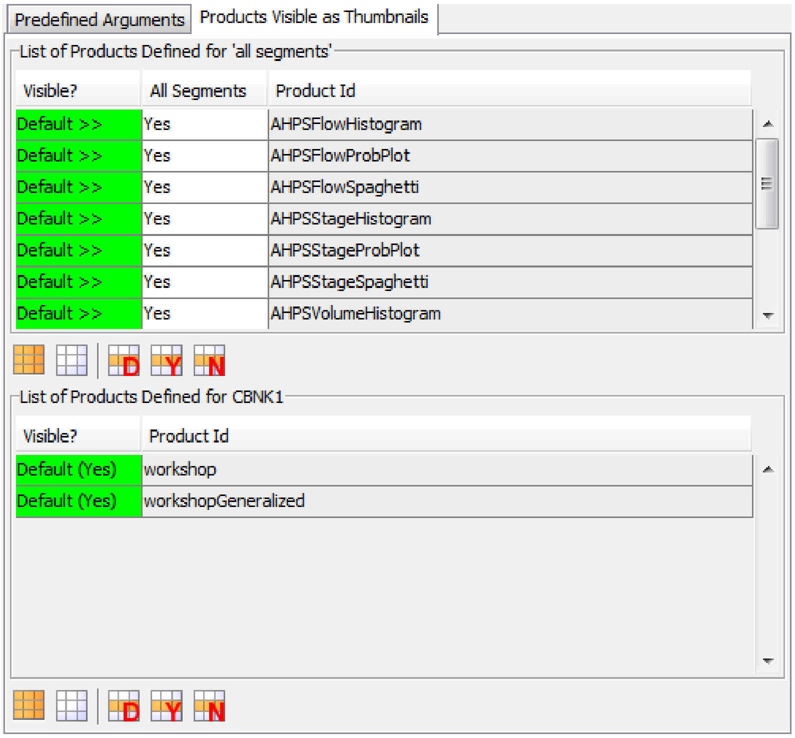
Some Graphics Generator products defined to be used by "all segments" may not be truly applicable to all forecast segments; for example the AHPS stage products may not be applicable to segments with no rating curve. In such cases, the products can be set to not be included in the **GraphGen Thumbnails Panel** for that segment. The steps are described below. Prior to performing those steps, the user must start CHPS. After starting CHPS, to make products not visible for a segment, do the following:

#	Action
1	Click on the GraphGen Tree (tab) on the left to open the **GraphGen Tree Panel**.
2	Click on the **Change Default Settings Button**, , in the toolbar of the **GraphGen Tree Panel**:  The **Modify Settings Dialog** will open (the exact appearance of this panel will vary based on what settings have already been modified):  Modify Settings Segments with Defined Settings all segments Predefined Arguments Products Visible as Thumbnails List of Predefined Arguments for all segments Argument Name Default Value Add Copy Remove Add Argument OK Cancel

<table>
<tr><th>#</th><th>Action</th></tr>
<tr><td>3</td><td>If the segment for which to make the change is listed in the <b>Segments With Defined Settings List</b>, select the segment by clicking on it. If not, then click on the <b>Add Button</b> under the list


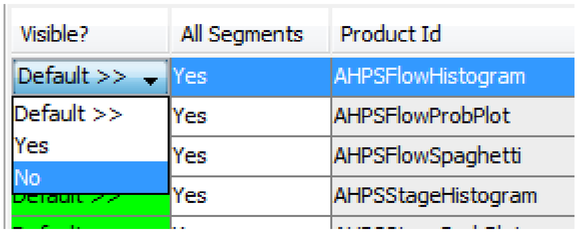
When the <b>Specify Segment Id Dialog</b> opens,


Enter Segment Id: CBNK1
,
enter segment id and click <b>OK</b>. The segment will appear in the <b>Segments With Defined Settings List</b>:


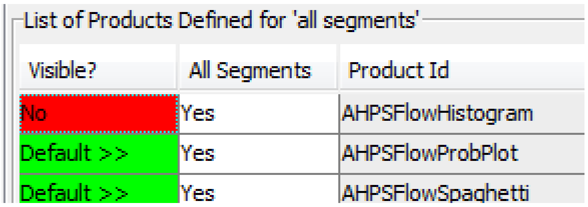
all segments
CBNK1</td></tr>
<tr><td>4</td><td>Select the <b>Products Visible as Thumbnails Tab</b> to make the <b>Products Visible as Thumbnails Panel</b> active:


List of Products Defined for 'all segments'
<table>
<tr><th>Visible?</th><th>All Segments</th><th>Product Id</th></tr>
<tr><td>Default &gt;&gt;</td><td>Yes</td><td>AHPSFlowHistogram</td></tr>
<tr><td>Default &gt;&gt;</td><td>Yes</td><td>AHPSFlowProbPlot</td></tr>
<tr><td>Default &gt;&gt;</td><td>Yes</td><td>AHPSFlowSpaghetti</td></tr>
<tr><td>Default &gt;&gt;</td><td>Yes</td><td>AHPSStageHistogram</td></tr>
<tr><td>Default &gt;&gt;</td><td>Yes</td><td>AHPSStageProbPlot</td></tr>
<tr><td>Default &gt;&gt;</td><td>Yes</td><td>AHPSStageSpaghetti</td></tr>
<tr><td>Default &gt;&gt;</td><td>Yes</td><td>AHPSVolumeHistogram</td></tr>
</table>
List of Products Defined for CBNK1
<table>
<tr><th>Visible?</th><th>Product Id</th></tr>
<tr><td>Default (Yes)</td><td>workshop</td></tr>
<tr><td>Default (Yes)</td><td>workshopGeneralized</td></tr>
</table>
Note that the top table displays the visibility status of products defined for ‘all segments’, such as the AHPS products, while the bottom displays only those specific to the segment.</td></tr>
</table>

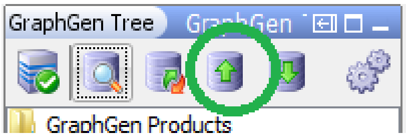
<table>
<tr><th>#</th><th>Action</th></tr>
<tr><td>5</td><td>Identify the row corresponding to the product to be made not visible, either defined for ‘all segments’ (top table) or defined for the segment (bottom table). Click on the ‘Visible?’ column for that row and select ‘No’:


<table>
<tr><th>Visible?</th><th>All Segments</th><th>Product Id</th></tr>
<tr><td>Default >></td><td>Yes</td><td>AHPSFlowHistogram</td></tr>
<tr><td></td><td>Yes</td><td>AHPSFlowProbPlot</td></tr>
<tr><td></td><td>Yes</td><td>AHPSFlowSpaghetti</td></tr>
<tr><td>Default >></td><td>Yes</td><td>AHPSStageHistogram</td></tr>
</table>

The background of the cell will change to red, indicating that the product is not visible:


List of Products Defined for 'all segments'
<table>
<tr><th>Visible?</th><th>All Segments</th><th>Product Id</th></tr>
<tr><td>🟥 No</td><td>Yes</td><td>AHPSFlowHistogram</td></tr>
<tr><td>🟩 Default >></td><td>Yes</td><td>AHPSFlowProbPlot</td></tr>
<tr><td>🟩 Default >></td><td>Yes</td><td>AHPSFlowSpaghetti</td></tr>
</table>

The ‘Visible?’ column indicates the visibility of a product by color: green for visible, red for not visible. The possible values are ‘Yes’, ‘No’, and ‘Default’. Default is only available for products defined for ‘all segments’. In such cases, the default product visibility is defined via the <b>Products Visible as Thumbnails Panel</b> for ‘all segments’. So, to make a product not visible by default (for all segments), select ‘all segments’ from the <b>Segments With Defined Settings List</b> and edit the <b>Products Visible as Thumbnails Panel</b> appropriately.</td></tr>
<tr><td>6</td><td>Click on the <b>OK Button</b> at the bottom of the <b>Modify Settings Dialog</b> to save the changes. If the <b>GraphGen Thumbnails Panel</b> is already open for the segment id for which a product was made not visible, then it will be redrawn removing the product’s thumbnail.</td></tr>
<tr><td>7</td><td>Make the change permanent by clicking on the <b>Upload Products Button</b> in the <b>GraphGen Tree Panel</b>:



When the confirmation dialog opens, click <b>Yes</b>.</td></tr>
</table>

2.7 *Porting Products (Templates) and Settings between SAs, OCs, and RFCs*

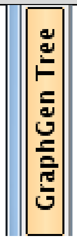
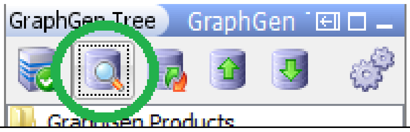
Porting products (templates) and settings, or copying them from one local data store to another, can be necessary in three cases:

- From SA to OC for adding the products to the central server.
- From SA to SA, for adding the products created in one stand-alone to another.
- Between RFCs, for dissemination of products between RFCs.

In all such cases, the following process can be followed. Before performing the steps below, start the CHPS session (SA or OC) where the products can be initially found. The process requires two steps: export products and settings from the source and import them into the target.

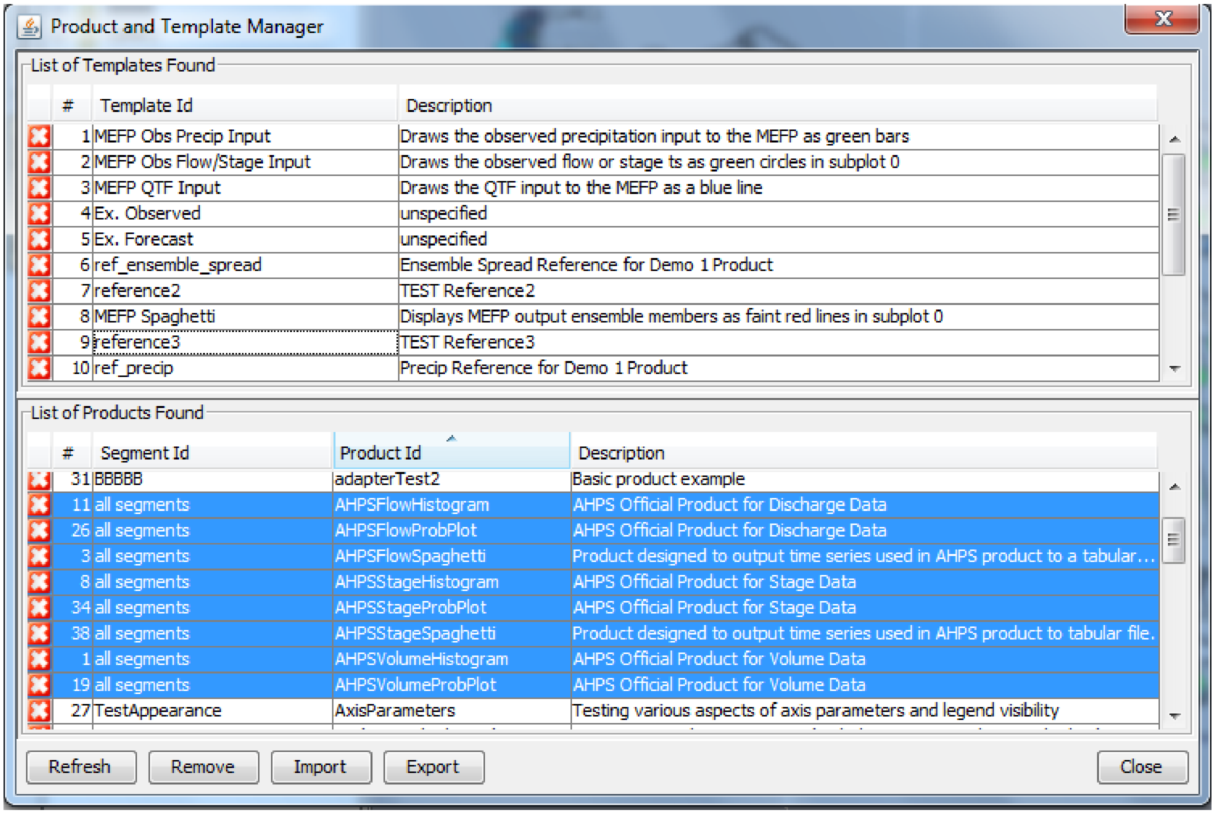
2.7.1 Exporting Products (Templates) and Settings

To export the products and settings, do the following:

#	Action
1	Click on the  (tab) on the left to open the **GraphGen Tree Panel**.
2	Click on the **Manage Products Button** to open the **Product and Template Manager Dialog**: 

**Action**

3 From both the **List of Templates Found** and the **List of Products Found**, select the templates and products to be ported. If all products and templates are to be ported, select no rows in either table. For example, to export the AHPS products delivered with Graphics Generator, make the following selections:



Product and Template Manager

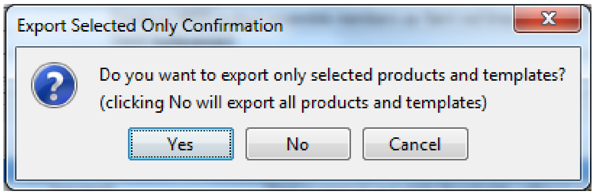
List of Templates Found

#	Template Id	Description
1	MEFP Obs Precip Input	Draws the observed precipitation input to the MEFP as green bars
2	MEFP Obs Flow/Stage Input	Draws the observed flow or stage ts as green circles in subplot 0
3	MEFP QTF Input	Draws the QTF input to the MEFP as a blue line
4	Ex. Observed	unspecified
5	Ex. Forecast	unspecified
6	ref_ensemble_spread	Ensemble Spread Reference for Demo 1 Product
7	reference2	TEST Reference2
8	MEFP Spaghetti	Displays MEFP output ensemble members as faint red lines in subplot 0
9	reference3	TEST Reference3
10	ref_precip	Precip Reference for Demo 1 Product

List of Products Found

#	Segment Id	Product Id	Description
31	BBBBB	adapterTest2	Basic product example
11	all segments	AHPSFlowHistogram	AHPS Official Product for Discharge Data
26	all segments	AHPSFlowProbPlot	AHPS Official Product for Discharge Data
3	all segments	AHPSFlowSpaghetti	Product designed to output time series used in AHPS product to a tabular…
8	all segments	AHPSStageHistogram	AHPS Official Product for Stage Data
34	all segments	AHPSStageProbPlot	AHPS Official Product for Stage Data
38	all segments	AHPSStageSpaghetti	Product designed to output time series used in AHPS product to tabular file.
1	all segments	AHPSVolumeHistogram	AHPS Official Product for Volume Data
19	all segments	AHPSVolumeProbPlot	AHPS Official Product for Volume Data
27	TestAppearance	AxisParameters	Testing various aspects of axis parameters and legend visibility

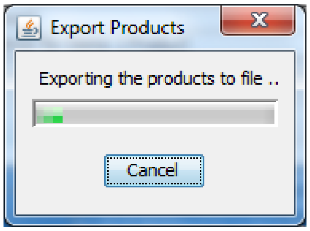
4 Click on the **Export Button**. If selections were made, when the **Export Selected Only Confirmation Dialog** opens, click **Yes** to only exported selected products and templates or **No** to export everything:



Export Selected Only Confirmation

Do you want to export only selected products and templates?
(clicking No will export all products and templates)

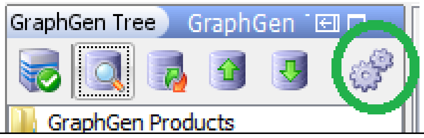
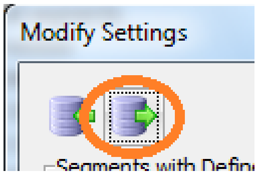
In the file dialog that opens, specify the file name to use and click **Open**. A status dialog will be displayed showing the progress of the export:



Export Products

Exporting the products to file ..

Once the import is complete, the dialog will close.

#	Action
5	Click close in the **Product and Template Manager Dialog** to close it.
6	Only perform Steps 5-7 if you have reason to deliver the settings associated with the products to port. When porting to another RFC, for instance, it is usually easiest to send instructions to the RFC for how to configure the Graphics Generator settings for the products to port. If it is decided that the settings will not be ported, go to Step 8. Click on the **Change Default Settings Button** to open the **Modify Settings Dialog**: 
7	In the **Modify Settings Dialog**, click on the **Export Settings Button**:  In the file dialog that opens, specify the file name to use and click **Open**. All settings specified in the **Modify Settings Dialog** will be exported to the file**.**

#	Action
8	This step should be skipped if it is okay to export all settings, which will often be the case when porting products within an RFC (from SA to SA or SA to OC). If it is not okay to port all settings, then open the file specified in the previous step and remove any settings that do not need to be ported. The file is an XML file; for example:

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<graphicsGeneratorSettings>
  <segmentSettingsList>
    <segmentSettings segmentId="all segments">
      <defaultArguments>
        <argument id="ahpsDisplayTimeZone">CST/CDT</argument>
        <argument id="ahpsMajorFloodingThresholdId">MAJOR</argument>
        <argument id="ahpsMinorFloodingThresholdId">FS</argument>
        <argument id="ahpsModerateFloodingThresholdId">MOD</argument>
      </defaultArguments>
      <allProductsVisibilityInThumbnails>
        <productId visible="false">MEFP Results</productId>
        <productId visible="false">MEFP-based ESP Output</productId>
        <productId visible="false">MEFP-based EnsPost Output</productId>
      </allProductsVisibilityInThumbnails>
    </segmentSettings>
    <segmentSettings segmentId="CBNK1">
      <allProductsVisibilityInThumbnails>
        <productId visible="true">MEFP Results</productId>
        <productId visible="true">MEFP-based ESP Output</productId>
        <productId visible="true">MEFP-based EnsPost Output</productId>
      </allProductsVisibilityInThumbnails>
    </segmentSettings>
  </segmentSettingsList>
</graphicsGeneratorSettings>
```

General rules-of-thumb for modifying the file are as follows:

a. If the products are to be ported to another RFC, remove the segment specific settings, shown in blue in the above example.

b. From the <defaultArguments> portions of the file, shown in yellow in the above example, remove any arguments that are not applicable to the ported product templates.

c. From the<allProductsVisibilityInThumbnails> portions of the file, shown in green in the above example, remove elements for any product templates not to be ported.

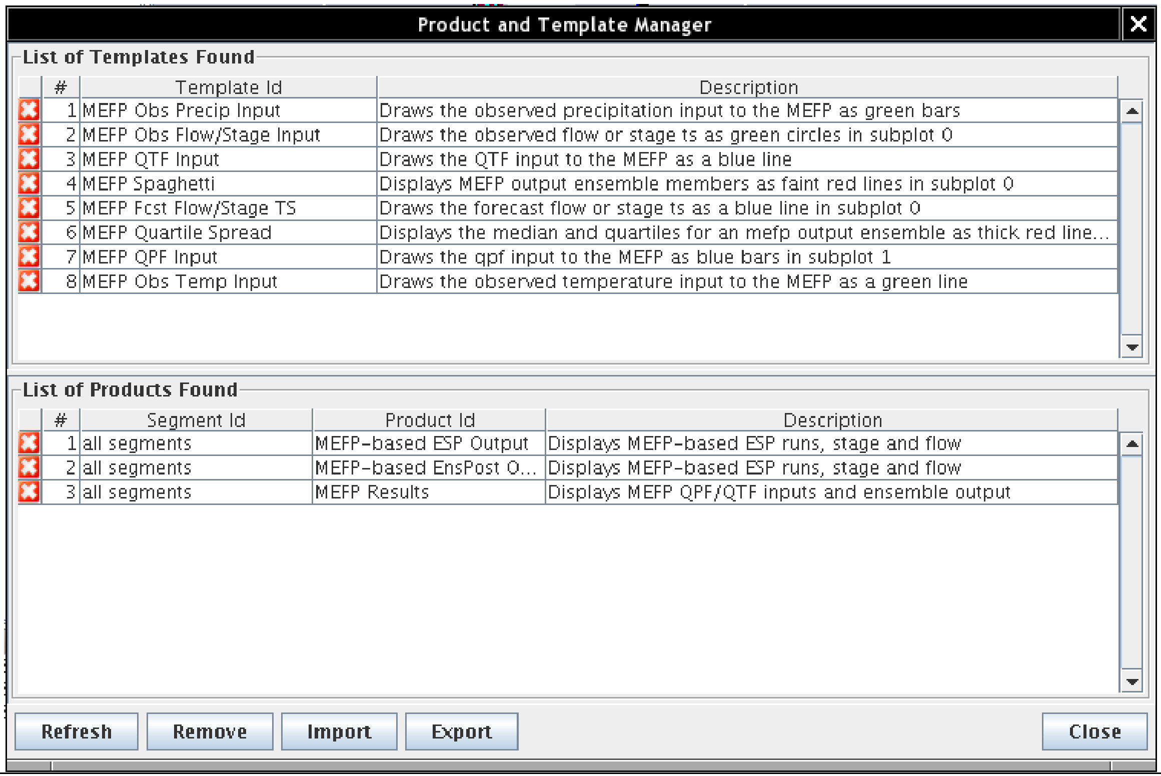
For example, if we want to port the AHPS products, which do not include any products with ids “MEFP…”, but do make use of the “ahps…” arguments, to another RFC, the file would be modified to the following:

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<graphicsGeneratorSettings>
  <segmentSettingsList>
    <segmentSettings segmentId="all segments">
      <defaultArguments>
        <argument id="ahpsDisplayTimeZone">CST/CDT</argument>
        <argument id="ahpsMajorFloodingThresholdId">MAJOR</argument>
        <argument id="ahpsMinorFloodingThresholdId">FS</argument>
        <argument id="ahpsModerateFloodingThresholdId">MOD</argument>
      </defaultArguments>
      <allProductsVisibilityInThumbnails>
      </allProductsVisibilityInThumbnails>
    </segmentSettings>
   </segmentSettingsList>
</graphicsGeneratorSettings>
```

#	Action
8	Click **OK** to close the **Modify Settings Dialog** and close this CHPS session.

2.7.2 Importing Products and Settings

To import the products and settings, open the target CHPS session, after starting the CHPS session, do the following:

#	Action
1	Click on the GraphGen Tree (tab) on the left to open the **GraphGen Tree Panel**.
2	Click on the **Manage Products Button**, 🔍, in the toolbar of the **GraphGen Tree Panel** to open the **Product and Template Manager Dialog** (exact content of this dialog depends what products and templates already exist) 

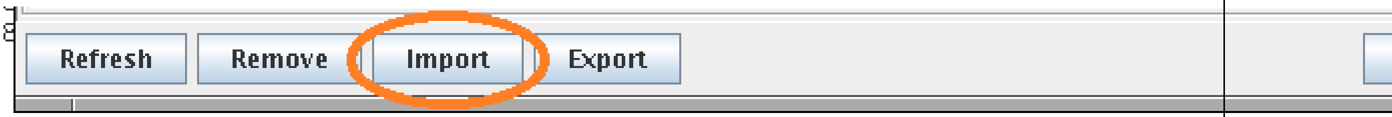
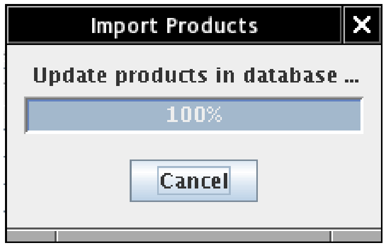
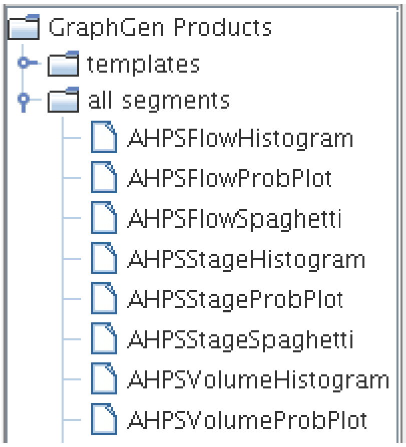
Product and Template Manager

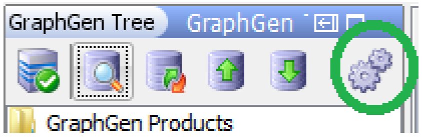
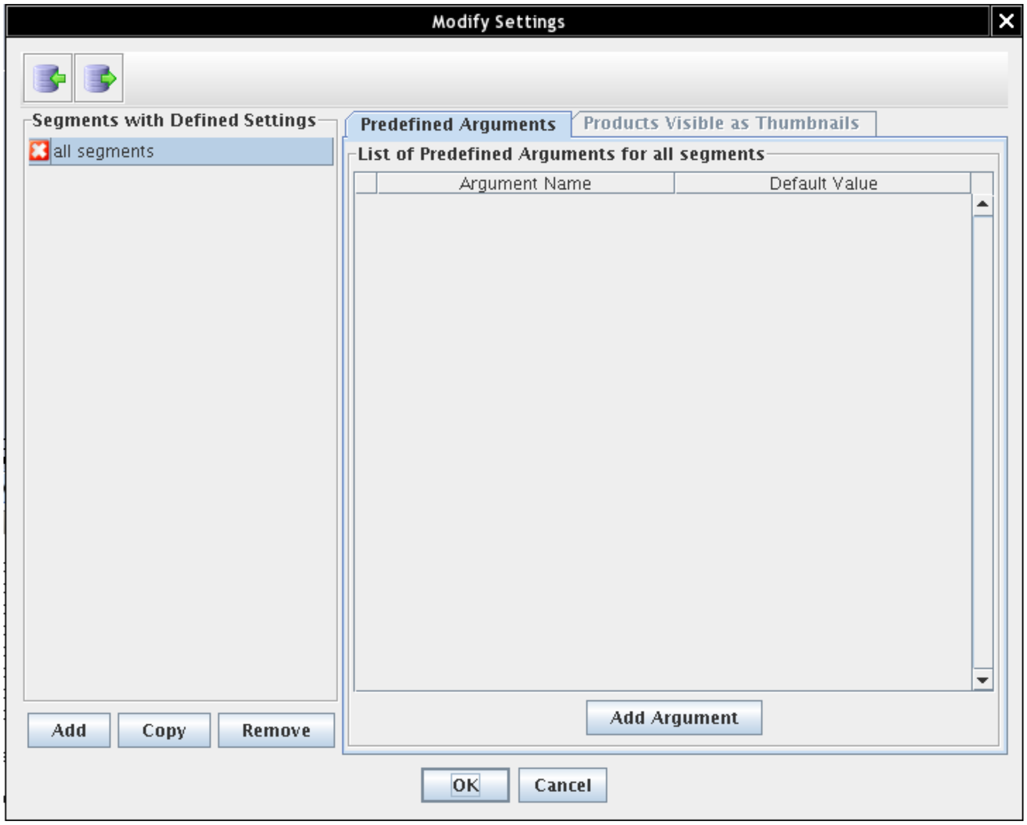
List of Templates Found

	#	Template Id	Description
☒	1	MEFP Obs Precip Input	Draws the observed precipitation input to the MEFP as green bars
☒	2	MEFP Obs Flow/Stage Input	Draws the observed flow or stage ts as green circles in subplot 0
☒	3	MEFP QTF Input	Draws the QTF input to the MEFP as a blue line
☒	4	MEFP Spaghetti	Displays MEFP output ensemble members as faint red lines in subplot 0
☒	5	MEFP Fcst Flow/Stage TS	Draws the forecast flow or stage ts as a blue line in subplot 0
☒	6	MEFP Quartile Spread	Displays the median and quartiles for an mefp output ensemble as thick red line...
☒	7	MEFP QPF Input	Draws the qpf input to the MEFP as blue bars in subplot 1
☒	8	MEFP Obs Temp Input	Draws the observed temperature input to the MEFP as a green line

List of Products Found

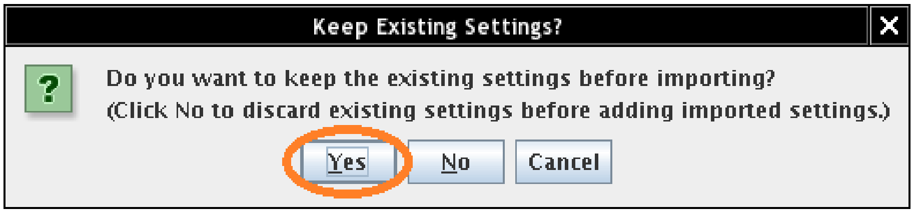
	#	Segment Id	Product Id	Description
☒	1	all segments	MEFP-based ESP Output	Displays MEFP-based ESP runs, stage and flow
☒	2	all segments	MEFP-based EnsPost O...	Displays MEFP-based ESP runs, stage and flow
☒	3	all segments	MEFP Results	Displays MEFP QPF/QTF inputs and ensemble output

#	Action
3	Click on the **Import Button** at the bottom of the dialog: 
4	In the **Select Import File Dialog** that opens, select the file created to contain the products and click **Open**. A status dialog will open displaying progress of the import:  Import Products Update products in database ... 100% Once the import is complete, the dialog will close and the imported products will be added to the **GraphGen Tree Panel**. Click on the **Close Button** in the **Product and Template Manager Dialog**. In the **GraphGen Tree Panel**, expand the branches for any segments for which products were ported, possibly including "templates" and "all segments", to see a listing of the products. For example, if the AHPS products are ported:  GraphGen Products templates all segments AHPSFlowHistogram AHPSFlowProbPlot AHPSFlowSpaghetti AHPSStageHistogram AHPSStageProbPlot AHPSStageSpaghetti AHPSVolumeHistogram AHPSVolumeProbPlot

#	Action
5	If product settings are to be ported as well, perform Steps 5-9. Otherwise, skip to Step 10. Click on the **Change Default Settings Button**, , in the toolbar of the **GraphGen Tree Panel**:  The **Modify Settings Dialog** will open (exact contents of this dialog depends on the settings that are already defined in the local data store):  Modify Settings Segments with Defined Settings all segments Predefined Arguments Products Visible as Thumbnails List of Predefined Arguments for all segments Argument Name Default Value Add Argument Add Copy Remove OK Cancel

<table>
<tr><th>#</th><th>Action</th></tr>
<tr><td>6</td><td>

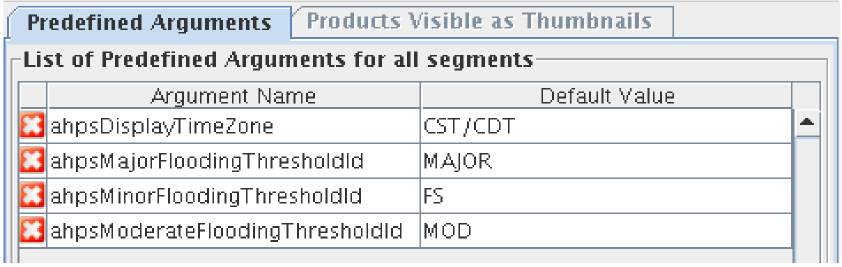
Click on the **Import Settings Button**, , in the tool bar of the dialog. When the **Select Import File Dialog** opens, select the ported settings file and click **Open**. When the **Keep Existing Settings? Dialog** opens, click **Yes** so that existing settings are not overwritten (it is only okay to click No if all products are being ported and the ported settings file is a complete copy of all settings that should be used):



Keep Existing Settings?

Do you want to keep the existing settings before importing?
(Click No to discard existing settings before adding imported settings.)

The **Predefined Arguments Tabbed Panel** will display the imported default predefined arguments settings for “all segments” (the list item selected initially in the **Segments with Defined Settings List** on the left side of the dialog). For example, for the AHPS products:

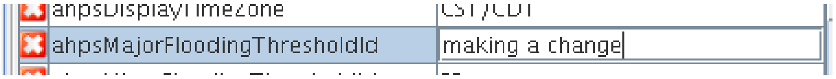


List of Predefined Arguments for all segments

<table>
<tr><th>Argument Name</th><th>Default Value</th></tr>
<tr><td>ahpsDisplayTimeZone</td><td>CST/CDT</td></tr>
<tr><td>ahpsMajorFloodingThresholdId</td><td>MAJOR</td></tr>
<tr><td>ahpsMinorFloodingThresholdId</td><td>FS</td></tr>
<tr><td>ahpsModerateFloodingThresholdId</td><td>MOD</td></tr>
</table>

</td></tr>
<tr><td>7</td><td>

Modify the default predefined arguments as needed and appropriate for the RFC. This should only be necessary when porting products between RFCs. To change a setting, click on the cell in the Default Value column and type the new value:

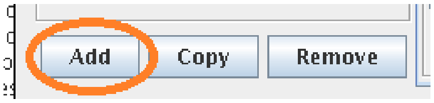
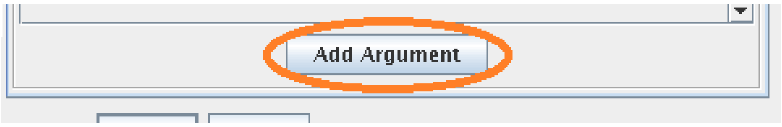
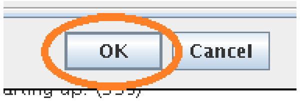
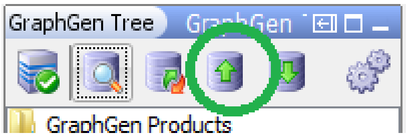


<table>
<tr><td>ahpsDisplayTimeZone</td><td>CST/CDT</td></tr>
<tr><td>ahpsMajorFloodingThresholdId</td><td>making a change</td></tr>
</table>

For example, for the AHPS products, it may be necessary to modify the time zone specified by the ahpsDisplayTimeZone predefined argument.

If modifications must be made, it is up to the user who exported the products from the source to specify appropriate instructions for whoever will import the products for the target.

</td></tr>
</table>

#	Action
8	Override the predefined arguments for segments that must use argument values that do not match the defaults set in the preceding step for “all segments”. For example, if the time zone for a segment does not match the default defined for “all segments”, it can be overridden. To Override a predefined argument, if the segment id (based on Topology.xml) of the segment for which to override settings is already in the **Segments with Defined Settings List**, select it. Otherwise click the **Add Button** under the list  and in the **Specify Segment Dialog** that opens enter the segment id and click **OK**. After the dialog closes, the segment will be selected in the **Segments with Defined Settings List**. With the appropriate segment selected, the **Predefined Arguments Tabbed Panel** on the left will display the overridden predefined arguments for that segment. If the argument to override is already in the **List of Predefined Arguments Table**, change its value if necessary (click on the Default Value cell, type in the new value, and press <Enter>). Otherwise, add the argument by clicking on the **Add Argument Button**  When the **Specify Argument Name Dialog** opens, set the argument name appropriately and click **OK**. Again, change the argument’s value by clicking on the Default Value cell in the **List of Predefined Arguments Table**, typing in the appropriate value, and pressing <Enter>.
9	Click **OK** to close the **Modify Settings Dialog**: 
10	Save all changes to the localDataStore. In the **GraphGen Tree Panel**, click on the **Upload Products Button**:  When the confirmation dialog opens, click **Yes**.

Upon completion of the steps above, the products and settings will be visible on the target CHPS session and should be confirmed by viewing the products via the **GraphGen Thumbnails Panel** and **GraphGen Viewer Panel**.

2.8 *General Tips for CHPS Configuration to Make Life Easier for Graphics Generator*

The following tips are provided for the general CHPS configuration files to make it easier to work with Graphics Generator products. If these tips are employed at your RFC, the amount of work to configure the FEWS PI-service and to setup the predefined arguments for Graphics Generator products via the **Modify Settings Dialog** will be minimized.

First, these tips are related to the following constraints of Graphics Generator:

- When querying the FEWS PI-service, Graphics Generator *must* provide a locationId or the PI-service will potentially return too much data, slowing down product generation and causing memory errors.
- The only CHPS id that Graphics Generator has access to as the active segment id, which it acquires through the FEWS PI-service.
- Thresholds are acquired from time series that are loaded through input series providers. The input series providers for a product are setup in the **Input Series Panel** of the **GraphGen Editor Panel**.

With these constraints in mind, the following configuration tips are recommended:

Again, the tips below are only recommendations intended to make it easier to setup Graphics Generator products. It is still possible to setup products without following these recommendations, but it will require more configuration and, therefore, will require more time and be more prone to errors.

- *Recommendation: Always use one locationId for all model output related to a segment (see Exceptions below).* For example, ADJUSTQ, ADDSUB, SAC-SMA, SNOW17, etc. models should all generate time series with the same locationId. Additionally, this locationId should match the id of the segment, if reasonable.

  From Graphics Generator's point of view, it only knows the active segment id, but default. Thus, products for a segment tend to be created to assume the predefined argument "defaultLocationId" (which is set to the active segment id, by default) indicates the locationId of <u>all</u> time series acquired. If time series have locationIds that vary, then either the "defaultLocationId" must be overridden (for example, see Section 2.5 of the *Graphics Generator AHPS Products Installation Guide*) or an id-mapping must be defined (for example, see Section 2.1.2 of the same guide).

  In practice, when configuring modules, there is no need to use different locationIds, because timeSeriesSet XML elements must also include a moduleInstanceId (or moduleInstanceSetId) and parameterId, and the combination of moduleInstanceId, locationId, and parameterId should be enough to indicate the type of data involved and its source. Furthermore, when viewing products through CHPS, the locationId should also not need to be model specific: the **Database Viewer** will display the moduleInstanceId and parameterId of time series, while plots should include titles and labels that are informative as to the source and type of data.

Exceptions:

- This cannot be done if one segment includes multiple gages for which streamflow, river stage, or reservoir time series must be generated. For such segments, each gage will likely require its own locationId. Graphics Generator predefined arguments and products may require customization to handle this (e.g., the defaultLocationId for a segment may need to be overridden to point to the correct gage and products may need to be copied to allow for multiple products for one segment).
- This does not apply to catchments for which precipitation and temperature time series are generated. Products that display catchment data will need to use special catchment ids in order to account for the possibility of a segment including multiple catchments.

- *Recommendation: Always define thresholds with ids that are not data type dependent.* For example, instead of "Record_Flow" and "Record_Stage", just define "Record". The ThresholdValueSets.xml configuration includes a unit that indicates the type of data (stage or flow). By making it data type dependent, Graphics Generator will require segment-specific overrides of threshold id arguments. Usually one will be chosen as the default threshold id defined for "all segments" (e.g., "Record_Flow"). That setting will then need to be overridden for each segment for which the threshold id is different (e.g., "Record_Stage").

- *Recommendation: Do not use qualifierIds unless absolutely necessary*. If the qualifierId is used only to indicate information that is otherwise obvious from the moduleInstanceId, locationId, and parameterId, then using qualifierIds only adds to the amount of work required to configure timeSeriesSet XML elements that use those time series, such as those in the FEWS PI-service configuration file. For Graphics Generator, this is mainly an inconvenience if some of the PI-service queries return time series with different qualifierIds, thus requiring multiple timeSeriesSet XML elements per query (see Section 2.1.1).

By following the tips above, the amount of work necessary to setup Graphics Generator products delivered as part of an OHD software release will be minimized. Furthermore, it will allow OHD to more easily enhance and develop software in the future, as it will lead to more configuration consistency across RFCs.

2.9 *Displaying HS Ensembles in AHPS Probability Plots*

Unlike the ESPADPInputGenerator that cuts one long time series into ensembles itself, Graphics Generator requires that those historical simulation (HS) ensembles have already been created in order for them to be used in creating the AHPS probability plots: AHPSFlowProbPlot, AHPSStageProbPlot and AHPSVolumeProbPlot. If Graphics Generator cannot find the historical ensembles, it will print error messages; for example:

```
"In ProbabilityPlot calculator, the computation date Wed Jun 19 12:00:00 GMT
2013 yielded an empty distribution/exceedance function; it may be outside the
start/end time of the data."
```

The HS ensembles can be created by using the sample transformation from an existing long historical simulation within CHPS. The new sample transformation should be run once per day alongside the standard CS ESP runs to create the comparable HS ensembles. The new transformation module looks very similar to the standard MAP SampleESP module. Here we provide general guidelines to create such a module and the corresponding workflow.

2.9.1 Create the Sample Transformation Module for Historical Ensembles

You can use the existing standard MAP SampleESP module configuration as the template. Following is an example of the sample transformation for historical ensemble at NERFC, Hudson_SampleESP_Historical.xml. We use the forecast group, Hudson, as the example.

```
<?xml version="1.0" encoding="UTF-8"?>
<transformationModule xmlns="http://www.wldelft.nl/fews" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.wldelft.nl/fews http://chps1/schemas/transformationModule.xsd" version="1.0">
<variable>
  <variableId>SSTG_Historic</variableId>
  <timeSeriesSet>
    <moduleInstanceSetId>ALL_STAGEQ_Forecast</moduleInstanceSetId>
    <valueType>scalar</valueType>
    <parameterId>SSTG</parameterId>
    <locationSetId>Catchments_Hudson</locationSetId>
    <timeSeriesType>simulated forecasting</timeSeriesType>
    <timeStep unit="hour" multiplier="6"/>
    <relativeViewPeriod unit="day" start="-36500" end="0"/>
    <readWriteMode>read only</readWriteMode>
  </timeSeriesSet>
</variable>

<variable>
  <variableId>SSTG_HS</variableId>
  <timeSeriesSet>
    <moduleInstanceId>Hudson_SampleESP_Historical</moduleInstanceId>
    <valueType>scalar</valueType>
    <parameterId>SSTG</parameterId>
    <locationSetId>Catchments_Hudson</locationSetId>
    <timeSeriesType>simulated forecasting</timeSeriesType>
    <timeStep unit="hour" multiplier="6"/>
    <relativeViewPeriod unit="day" start="0" end="0" endOverrulable="true"/>
    <readWriteMode>add originals</readWriteMode>
  <ensembleId>HS</ensembleId>
  </timeSeriesSet>
</variable>

<variable>
  <variableId>QINE_Historic</variableId>
  <timeSeriesSet>
    <moduleInstanceSetId>ADJUSTQ_Forecast</moduleInstanceSetId>
    <valueType>scalar</valueType>
    <parameterId>QINE</parameterId>
    <locationSetId>Catchments_Hudson</locationSetId>
    <timeSeriesType>simulated forecasting</timeSeriesType>
    <timeStep unit="hour" multiplier="6"/>
    <relativeViewPeriod unit="day" start="-36500" end="0"/>
    <readWriteMode>read only</readWriteMode>
  </timeSeriesSet>
</variable>

<variable>
  <variableId>QINE_HS</variableId>
  <timeSeriesSet>
    <moduleInstanceId>Hudson_SampleESP_Historical</moduleInstanceId>
    <valueType>scalar</valueType>
    <parameterId>QINE</parameterId>
    <locationSetId>Catchments_Hudson</locationSetId>
    <timeSeriesType>simulated forecasting</timeSeriesType>
    <timeStep unit="hour" multiplier="6"/>
    <relativeViewPeriod unit="day" start="0" end="0" endOverrulable="true"/>
    <readWriteMode>add originals</readWriteMode>
    <ensembleId>HS</ensembleId>
  </timeSeriesSet>
</variable>

<transformation id="Hudson_SampleHS_SSTG">
  <sample>
     <historical>
       <inputVariable>
           <variableId>SSTG_Historic</variableId>
       </inputVariable>
         <moduleParameterFile>Hudson_SampleESP_Forecast</moduleParameterFile>
       <outputVariable>
         <variableId>SSTG_HS</variableId>
       </outputVariable>
     </historical>
   </sample>
</transformation>

<transformation id="Hudson_SampleHS_QINE">
   <sample>
     <historical>
       <inputVariable>
          <variableId>QINE_Historic</variableId>
       </inputVariable>
         <moduleParameterFile>Hudson_SampleESP_Forecast</moduleParameterFile>
       <outputVariable>
```

```
      <variableId>QINE_HS</variableId>
     </outputVariable>
    </historical>
  </sample>
</transformation>
</transformationModule>
```

The Graphics Generator AHPS probability plots use two parameter Ids: SSTG and QINE. In the example, we define four variables, one input and one output for each of the parameters. You should modify the parts highlighted in yellow to your specific RFC specification. Please also note that the output variables, SSTG_HS and QINE_HS, must define an ensemble ID (highlighted in green). It is HS here. This ensemble ID must match the ensemble ID defined in the file …/Config/PiServiceConfig/GraphGen.xml for the same parameter.

Next, modify the file …/Config/RegionConfigFiles/ModuleInstanceDescriptors.xml to include the new sampling modules. For example,

```
  <moduleInstanceDescriptor id="Hudson_SampleESP_Historical">
      <moduleId>TransformationModule</moduleId>
  </moduleInstanceDescriptor>
```

2.9.2 Create the Sample Transformation Workflow

In the workflow, before executing the sampling transformation activity, two other activities, SetTimes_LastObserved, and SetTimes_ESP_Forecast, are executed first. The first module, SetTimes_LastObserved, interpolates the input MAT data. The second module, SetTimes_ESP_Forecast, is a FEWS forecastLengthEstimator. You should have these two modules already for the CS ESP simulations. Following is an example workflow configuration, Hudson_PostProcessing_ESP_Historical.xml.

```
<?xml version="1.0" encoding="UTF-8"?>
<workflow xmlns="http://www.wldelft.nl/fews" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.wldelft.nl/fews http://chps1/schemas/workflow.xsd" version="1.1">
<activity>
  <runIndependent>true</runIndependent>
  <moduleInstanceId>SetTimes_LastObserved</moduleInstanceId>
</activity>
<activity>
  <runIndependent>true</runIndependent>
  <moduleInstanceId>SetTimes_ESP_Forecast</moduleInstanceId>
</activity>
<activity>
  <runIndependent>true</runIndependent>
<moduleInstanceId>Hudson_SampleESP_Historical</moduleInstanceId>
</activity>
</workflow>
```

Change the moduleInstanceId marked in yellow . The SetTimes_ESP_Forecast module includes an the XML element <minForecastLength multiplier="100" unit="day"/> . The value defined here determines the sampling length. It has to match (be longer or equal to) the period required by the AHPS probability products (See Section 2.9.4).

Next, add the workflow to your …/Config/RegionConfigFiles/WorkflowDescriptors.xml file. For example,

```
<workflowDescriptor id="Hudson_PostProcessing_ESP_Historical" name="Hudson_PostProcessing_ESP_Historical"
forecast="true" visible="true" allowApprove="true" autoApprove="true"/>
```

Please note the workflow has to be approved before the historical ensembles can be retrieved successfully by Graphic Generator.

2.9.3 Modify the GraphGen configured query in the GraphGen.xml file

Check the <timeSeriesSet> defined for the queries with ids <id>SSTG HS<id> and <id>QINE HS<id>, to make sure it matches with the historical ensembles created by the sampling transformation. For example,

```
      <!--  Returns the historical ESP streamflow forecasts -->
  <timeSeries>
    <id>QINE HS</id>
    <timeSeriesSet>
      <moduleInstanceId>Hudson_SampleESP_Historical</moduleInstanceId>
      <valueType>scalar</valueType>
      <parameterId>QINE</parameterId>
      <locationId>MTRN6HUD</locationId>
      <timeSeriesType>simulated forecasting</timeSeriesType>
      <timeStep unit="hour" multiplier="6" timeZone="GMT-0" />
      <readWriteMode>read only</readWriteMode>
      <ensembleId>HS</ensembleId>
      <ensembleMemberIndexRange start="1952" />
    </timeSeriesSet>
  </timeSeries>

  <!--  Returns the historical ESP stage forecasts -->
  <timeSeries>
    <id>SSTG HS</id>
    <timeSeriesSet>
      <moduleInstanceId>Hudson_SampleESP_Historical</moduleInstanceId>
      <valueType>scalar</valueType>
      <parameterId>SSTG</parameterId>
      <locationId>MTRN6HUD</locationId>
      <timeSeriesType>simulated forecasting</timeSeriesType>
      <timeStep unit="hour" multiplier="6" timeZone="GMT-0" />
      <readWriteMode>read only</readWriteMode>
      <ensembleId>HS</ensembleId>
      <ensembleMemberIndexRange start="1952" />
    </timeSeriesSet>
  </timeSeries>
```

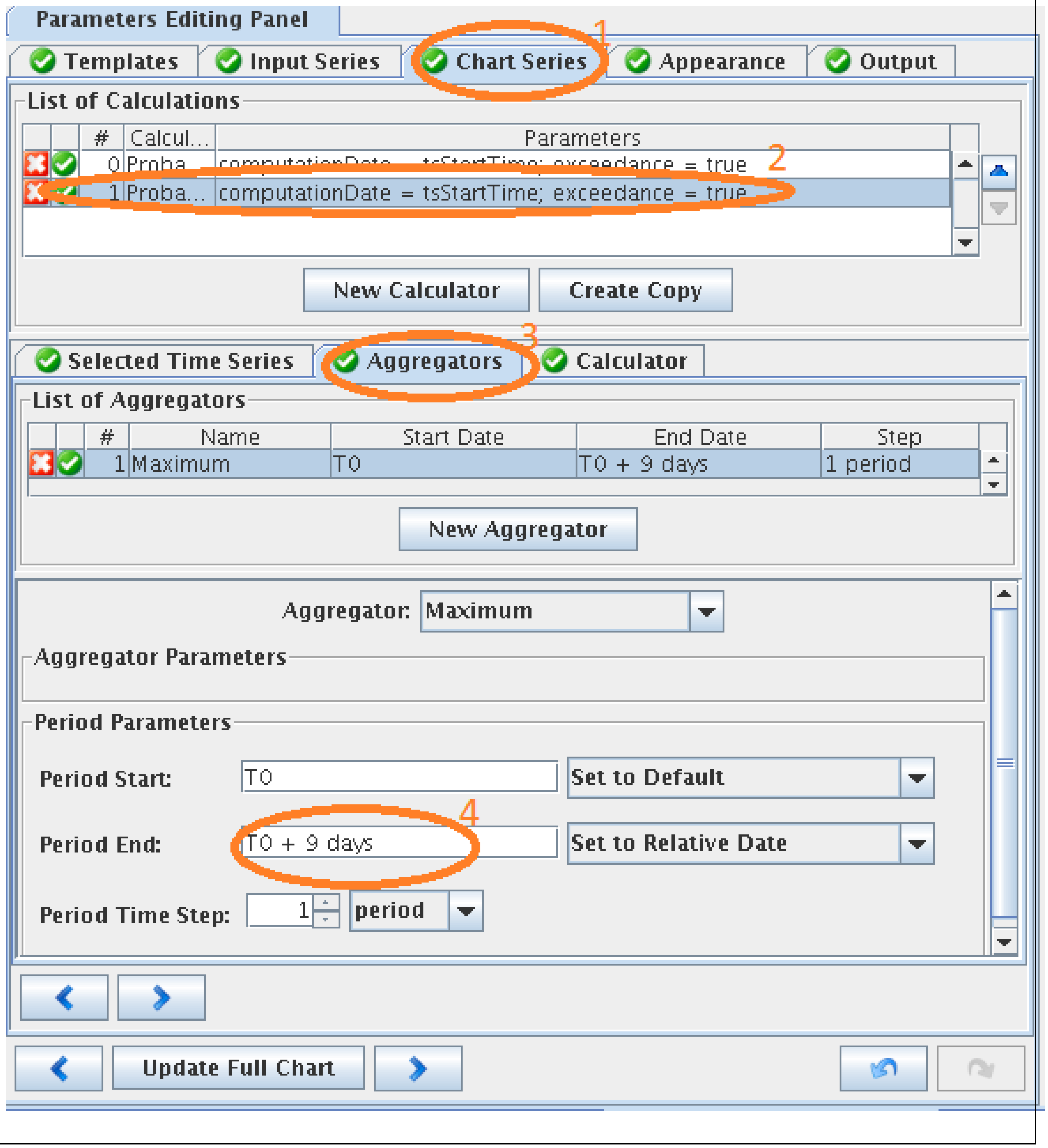
Note that the <moduleInstanceId> and the <ensembleId> match what have been defined in the sample transformation module.

2.9.4 Check the <minForecastLength> and the Period Required by the Probability Products

Open the SetTimes_ESP_Forecast module, examine the value of XML element minForecastLength. For example, <minForecastLength multiplier=“100” unit=”day”/>, defines the sampling length to be 100 days.

In CHPS, make sure the start period and end period defined in the **Aggregators Panel** matches the sampling length. Otherwise, the historical simulation will not be displayed in the plots.

#	Action
1	Start CHPS.
2	Open a probability product in the **GraphGen Editor**.

#	Action
3	1. Select **Chart Series** Tab. 2. In the **List of Calculations Table**, select the historical simulation calculations (the second row). 3. Select the **Aggregators** Tab below the **List of Calculations**. 4. Modify the **Period Parameters** to make sure the **Period Start** and **Period End** fields match the value defined in the `SetTimes_ESP_Forecast` module. The period length (end period – start period) used in aggregation must be less than or equal to the `minForecastLength` value defined in the `SetTimes_ESP_Forecast` module. 

List of Calculations

		#	Calcul...	Parameters
☒	✓	0	Proba	computationDate = tsStartTime; exceedance = true
☒	✓	1	Proba...	computationDate = tsStartTime; exceedance = true

List of Aggregators

		#	Name	Start Date	End Date	Step
☒	✓	1	Maximum	T0	T0 + 9 days	1 period

Aggregator: Maximum

Aggregator Parameters

Period Parameters

Period Start: T0 — Set to Default

Period End: T0 + 9 days — Set to Relative Date

Period Time Step: 1 period

2.9.5 Workflow Uniqueness Problem

It has been reported (Fogbugz 1083 and 1061) that when running multiple workflows for different forecasting groups as configured in section 2.9.2, the HS ensembles created by the last workflow will overwrite the HS ensembles of previous forecasting group. Therefore, the previous HS ensembles are not available for Graphics Generator to display the probability plots for the previous forecasting group. One solution is to group the multiple workflows for different forecasting groups into one single workflow that run all the forecasting group workflows. Please see the FogBugz 1083 and 1061 for details.

3 Trouble Shooting

3.1 *Time Series Cannot Be Acquired via the PI-service*

If the CHPS **Logs Panel** displays a message similar to the following when constructing a GraphGen product:

```
04-03-2013 11:34:09 WARN - Chart series #0 was calculated but is empty.
04-03-2013 11:34:09 WARN - No time series match selection parameters:
    [TimeSeriesSelectionParameters: xmlTagName = 'timeSeriesSelection'.]
```

Then it may be the case that time series that are expected for the product are not being returned by the FEWS PI-service. First identify the PI-service query that returns the time series. The query must be defined in a file under the directory

…/Config/PiServiceConfigFiles/.

Determine if time series <u>should</u> be found. For some products, it may be the case that some time series are optional, meaning that the product will still display if they are not found. For the AHPS products, each product is computed and constructed from a single ensemble and, if that ensemble is not available, then the product cannot be constructed and displayed. However, for other products, it may be the case that not all time series are required. For example, the products may display "optional" observed data as a frame of reference for the forecast data displayed, but the observed data may not always be available for all segments. In such a case, the forecast portion of the product will still be constructed and missing observed time series do not indicate an error.

If the missing time series is an error (i.e., the time series should be available, but are not properly loaded via the FEWS PI-service), then the error is likely one of the following:

- *The port number of the PI-service connection is wrong*. See Section 2.1.

- *The locationId of the time series acquired via the FEWS PI-service does not match the locationId expected by the Graphics Generator*. This problem is product specific, but typically the id will be either the segment id or defined by the Graphics Generator setting (predefined argument) "defaultLocationId". Make modifications as appropriate for the product, possibly defining an id-mapping to map the locationIds to the segment id (see Section 2.1.2 of the *Graphics Generator AHPS Products Installation Guide*, for example) or overriding the defaultLocationId for the affected segments (see Section 2.3, Steps 7 and 8 of the *Graphics Generator AHPS Products Installation Guide*).

  This problem is most easily detected by opening the **GraphicsGenerator Thumbnails Panel** for the segment in question and comparing it with output from the **Database Viewer.**

- *The identity mapping (see the terminology in Section 1.3) is missing from the id-mapping file defined in the applicable PI-service configuration file*. For example, for the default PI-service configuration file delivered with Graphics Generator, GraphGen.xml, the id-mapping file is the file Config/IdMapFiles/IdExportPiService.xml. This problem is product specific; make changes as appropriate for the product.

If no id-mappings are defined for a segment, the identity mapping is not necessary for that segment. Only include an identity mapping if other mappings are defined for that segment.

- *The workflow that generates the time series has not been run yet*.

- *A PI-service query involved returns a time series of type "simulated forecasting", but the workflow that generated the time series was not approved*. The FEWS PI-service only returns forecast time series that are approved. Other types of time series need not be approved.

- *The PI-service query is incorrect*. Use the CHPS **Database Viewer** to view the time series you believe should be acquired via the PI-service. Compare table fields with the queries defined in the appropriate PI-service configuration file, paying close attention to the following:

  - locationId
  - parameterId
  - qualifierId
  - timeStep
  - ensembleId
  - ensembleMemberIndex

  See Section 2.5 for testing the PI-service configuration via the **GraphGen Editor Panel**.

  Pay close attention to the log output in the CHPS **Logs Panel** when debugging problems in the configuration. For example, if the log output is similar to the following (note the highlighted text):

```
04-03-2013 11:34:09 WARN - Chart series #0 was calculated but is empty.
04-03-2013 11:34:09 WARN - No time series match selection parameters:
    [TimeSeriesSelectionParameters: xmlTagName = 'timeSeriesSelection'.]
04-03-2013 11:34:09 WARN - For client id HEFSGraphGen, query id FLOW RAW
    OUTPUT and location id FTSC1, and other parameters, acquired time series were all
    missing and will be discarded.
```

  then the configuration of the PI-service query is valid but did not yield any data. This is commonly caused by an incorrect timeStep (6-hours, but the data is 1-hour), a missing qualifierId in the query, a missing ensembleId, and so on.

On the other hand, if the log output is similar to the following (note the highlighted text):

```
04-01-2013 18:04:21 WARN - For client id HEFSGraphGen, query id MEFP QPF
    INPUT and location id DOSC1H, and other parameters, acquired no time series via
    the FEWS pi-service.
```

then the configuration of the PI-service query is invalid. This is commonly caused by a reference to an undefined moduleInstanceId or moduleInstanceSetId, an undefined locationId or locationSetId, an invalid timeStep, and so on.

If the problem persists after checking everything above, and making appropriate fixes, report the problem as a bug.

3.2 *Products Take a Long Time to Build*

If a product takes a long time to build, performance may be improved by increasing the time-step of the time series used as input to Graphics Generator in order to build the product. The following guidance is recommended:

> *If time series take in excess of one second to load, then see if the time step can be increased for the product being constructed by performing aggregations via CHPS transformations and using the aggregated time series as input to Graphics Generator.*

For example, since the AHPS products delivered with this release use 1-week aggregations or longer, if the time series have a one-hour time step, performance can be improved greatly by aggregating those time series to 6-hours or one-day, and using those time series as input to Graphics Generator for building AHPS products.

3.2.1 More on Time Spent Building Products

When viewing a product via the **GraphGen Viewer Panel**, time spent building the product is provided in messages displayed in the CHPS **Logs Panel**; for example:

```
11-05-2012 14:56:34 INFO - Successfully built JFreeChart in 0 msec.
11-05-2012 14:56:34 INFO - Building JFreeChart...
11-05-2012 14:56:34 INFO - Successfully prepared chart (time series load time: 796 msec;
prepare chart time: 140 msec).
11-05-2012 14:56:34 INFO - Chart preparation times: time spent building references = 0 msec;
time spent doing computations = 94 msec; time spent putting together chart = 46 msec).
11-05-2012 14:56:33 INFO - Preparing chart..
```

Similar log messages are provided in the diag.xml file output when generating products via a workflow. The total time spent building a product is the sum three component times:

- Time spent loading a time series, highlighted in yellow, above.
- Time spent computing chart series and preparing the chart, highlighted in green, above.
- Time spent rendering the chart image for display in either the CHPS interface or an image file (.png, .jpg, etc.), highlighted in blue, above (typically a very small number).

The time spent loading a time series is further broken down into two parts, both dependent on the FEWS PI-service:

- Time spent acquiring the raw time series via the FEWS PI-service (consists of acquiring XML and parsing that XML).
- Time spent converting the units of the raw time series to display units via the FEWS PI-service (the raw units are metric while the display units are usually English).

The time spent computing chart series and preparing the chart and the time spent rendering a chart image are outside the control of the user. However, those times typically are not the primary bottleneck. The time spent loading time series is the primary bottleneck usually accounting for 66% or more of the total time spent building a product. Hence the recommendation above to aggregated time series in CHPS.

3.3 *Products Cannot Be Found During Adapter Execution*

When executing the Graphics Generator Model Adapter, the following is an example of a message indicating that products could not be found:

```
<line level="3" description="Checking for existence of products and
   files..."/>
<line level="1" description="For segment 'CNNN6DEL', product matching
   'MEFP.*' does not exist."/>
<line level="1" description="For segment 'WALN6DEL', product matching
   'MEFP.*' does not exist."/>
```

This is likely due to one of the following problems:

- Error in the module configuration file. Double product specifying lines in the module configuration file exportRunFileActivity properties section. The property is described in model adapter section of the *Graphics Generator Reference Manual*.

- The wrong FEWS PI-service is being accessed. If this is a scheduled task, the back-end service must be accessed; see the *Graphics Generator Installation Guide*. If this uses a FEWS PI-service associated with a running CHPS session, then the port number may be wrong. Identify the correct port number for the CHPS session (see Section 2.3) and specify it in the adapter configuration file (see the *Graphics Generator Installation Guide* and the model adapter section of the *Graphics Generator Reference Manual*).

- The products needed for the run are not accessible via the used FEWS PI-service. Make sure that the FEWS PI-service is accessing a database in which the products exist. See Section 2.7 for instructions on porting products to the needed database, if necessary.

If the problem persists after checking everything above, report the problem as a bug.

3.4 *Thresholds are Not Displayed*

Graphics Generator products will often include thresholds that are defined within the configuration file

<*configuration_dir*>/RegionConfigFiles/ThresholdValueSets.xml

and associated with a time series that is provided to Graphics Generator via an input series provider that accesses the FEWS PI-service. For example, see the *Graphics Generator AHPS Products Installation Guide*. When a chart is correctly displayed, but thresholds that are expected to be displayed are not, it is likely due to one of the following problems:

1. **PROBLEM: ThresholdValueSets.xml is not configured correctly**.
   The problem may be that the expected threshold is not defined within the configuration file above or it is not associated with the provided time series. Graphics Generator accesses the thresholds via associated time series supplied via input series, so if threshold to display is not associated correctly, then the default threshold will not be found.
   SOLUTION: Correct the ThresholdValueSets.xml configuration file.

2. **PROBLEM: Improperly defined arguments**.
   If arguments are used to define threshold identifiers or values, such as the ahpsMinorFloodingThresholdId argument for the AHPS products, then it may be that the argument is defined incorrectly for the forecast segment/location of the product.
   SOLUTION: Correct the argument definition using the tools provided with Graphics Generator, including the **Modify Settings Dialog** (accessible via the **GraphGen Tree Panel** and **GraphGen Thumbnails Panel**) and the global arguments defined for the product (editable via the **GraphGen Editor Panel**).

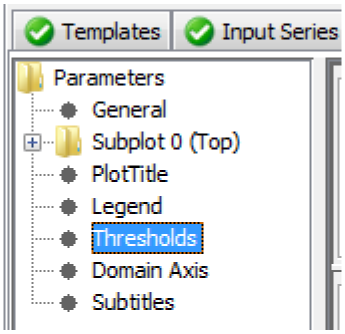
3. **PROBLEM: Incorrect Graphics Generator product definition**.
   The definition of the product, beyond the arguments, may be the source of the problem. For example:

   a. Not checking the **Include Thresholds Checkbox** for the appropriate input series provider. See *Graphics Generator Reference Manual* section on the **Input Series Panel**.
   b. Failing to load the time series based on the input series provider options. For example, the location id, parameter id, ensemble id, or another option of the input series provider may not be correctly specified. This may also lead to more general problems displaying the product. See *Graphics Generator Reference Manual* section on the **Input Series Panel**.
   c. Incorrectly referring to the default threshold within the parameters of the threshold. The parameters are set within the **Threshold Parameters Panel** of the **Appearance Panel**. See *Graphics Generator Reference Manual* section on the **Appearance Panel**.

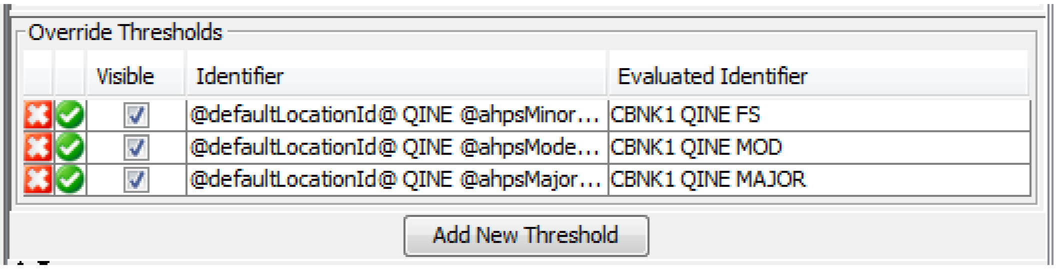
   SOLUTION: Correct the product definition as appropriate.

3.4.1 Diagnosing the Problem

Action: Select the appropriate forecast segment within the CHPS **Forecasts Panel**. Open the product in question in the **GraphGen Editor Panel**. Click on the **Appearance** tab to display the **Appearance Panel** and select the **Thresholds** item from the **Plug-in Tree** on the left:



Question 1: Are any override thresholds listed in the **Override Thresholds Table** of the **Threshold Parameters Panel**? For example:

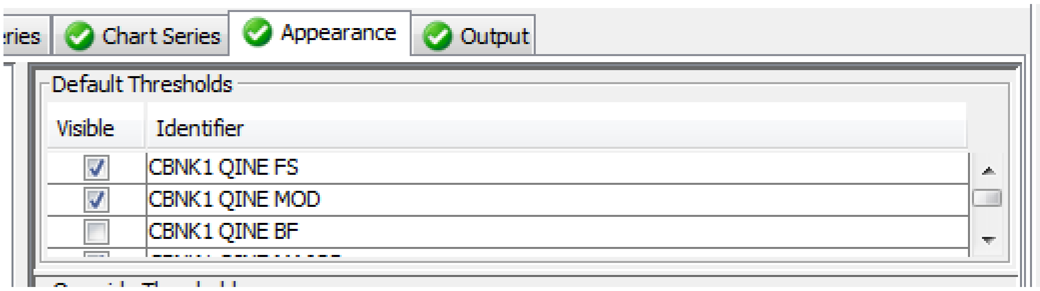


Override Thresholds

		Visible	Identifier	Evaluated Identifier
☒	✓	☑	@defaultLocationId@ QINE @ahpsMinor…	CBNK1 QINE FS
☒	✓	☑	@defaultLocationId@ QINE @ahpsMode…	CBNK1 QINE MOD
☒	✓	☑	@defaultLocationId@ QINE @ahpsMajor…	CBNK1 QINE MAJOR

If there are no override thresholds, then the product is not defined to display thresholds, hence there is no reason to go any further with diagnosing the problem.

Question 2: Do you see default thresholds supplied in **Default Thresholds Table** at the top? For example:



Default Thresholds

Visible	Identifier
☑	CBNK1 QINE FS
☑	CBNK1 QINE MOD
□	CBNK1 QINE BF

If you do not see any entries in this table, then the thresholds are not being found with the input time series.

Action: Switch to the **Input Series Panel** (click on the **Input Series** tab), select the appropriate input series provider from the **List of Input Series Providers Table** and verify the following:

In some cases, the thresholds may be provided via a referenced template listed within the **Templates Panel** in the **List of Referenced Templates Table**; thresholds loaded in referenced templates will also be visible in products that display those templates. Identify the template that loads the time series and open it in the **GraphGen Editor Panel**. Answer the questions below using that template, being sure to save any fixes made before switching to the original product to check if the fix worked.

1. The **Include Thresholds Checkbox** must be checked.

   NO: Check the checkbox. Return to the **Appearance Panel**: Are the default thresholds visible in the table? If not, return to the **Input Series Panel** and continue.

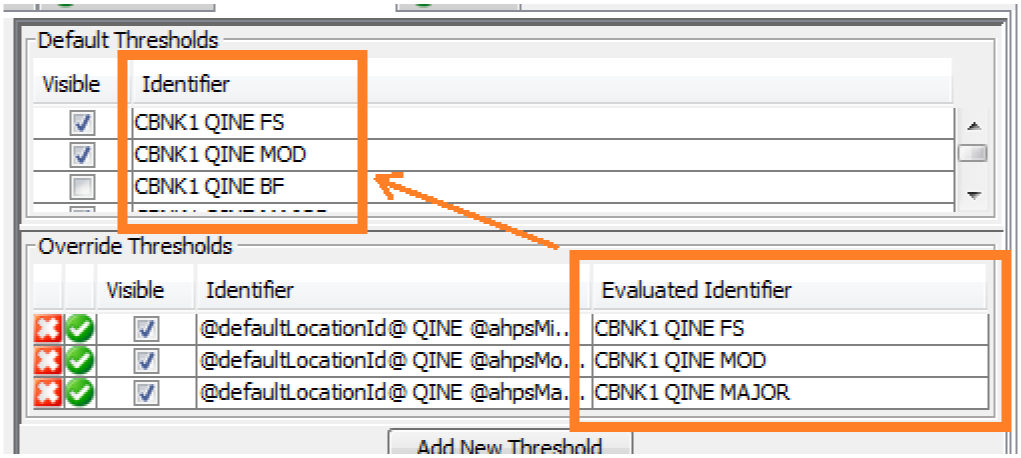
   YES: Go to the next item.

2. Time series are displayed in the **Current Status Display Panel** in the upper right of the **GraphGen Editor Panel**. This should be the case whenever the provider is selected within the **List of Input Series Providers Table** and the time series are successfully loaded.

   NO: The provider may not be configured correctly to load the expected time series. Either change the options of the **Provider Options Panel** or correct any involved configuration files, such as those under <*configuration_dir*>/PiServiceConfigFiles. Return to the **Appearance Panel**: Are the default thresholds visible in the table? If not, **Input Series Panel** and go to the next question.

   YES: The source time series do not contain the thresholds. If the input series provider uses the PI-service (i.e., providers ConfiguredQuery, SegmentTimeSeries, SelectTimeSeries), check the ThresholdValueSets.xml file and make sure the thresholds are associated with the time series acquired via the input series provider. Close and reopen the **GraphGen Editor** in order to ensure new data is loaded via the provider. Return to the **Appearance Panel**: Are the default thresholds visible in the table? If not, report this problem as a bug.

Question 3: Do the default thresholds that should be displayed have a corresponding row in the **Override Thresholds Table** for which the “Evaluated Identifier” column value matches the “Identifier” column value in the **Default Thresholds Table**? For example:



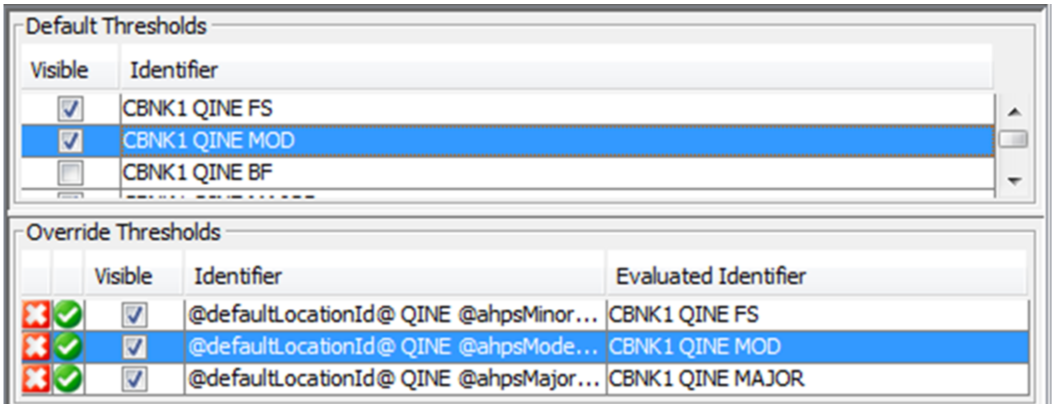
Default Thresholds

Visible	Identifier
☑	CBNK1 QINE FS
☑	CBNK1 QINE MOD
☐	CBNK1 QINE BF

Override Thresholds

		Visible	Identifier	Evaluated Identifier
☒	✓	☑	@defaultLocationId@ QINE @ahpsMi...	CBNK1 QINE FS
☒	✓	☑	@defaultLocationId@ QINE @ahpsMo...	CBNK1 QINE MOD
☒	✓	☑	@defaultLocationId@ QINE @ahpsMa...	CBNK1 QINE MAJOR

One way to check this is to select the row of the default threshold in the **Default Thresholds Table**. If a corresponding row of the **Override Thresholds Table** is defined, it will also be selected:



Default Thresholds

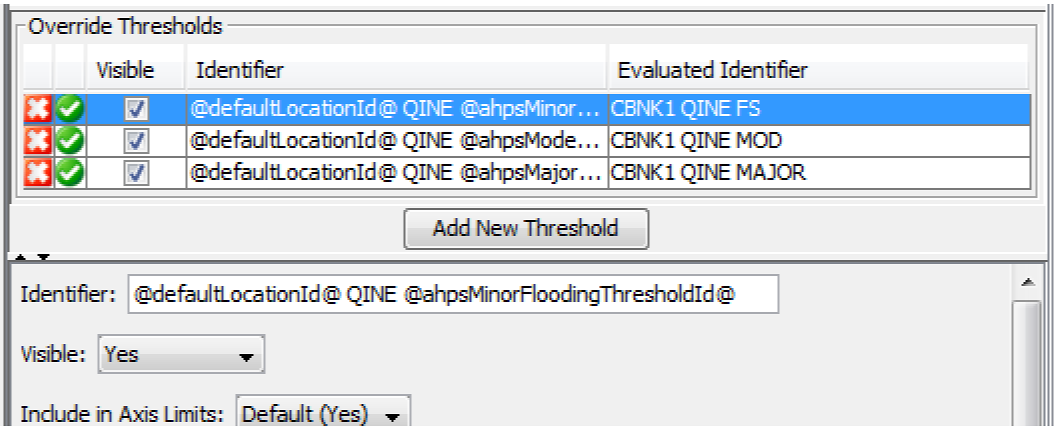
Visible	Identifier
☑	CBNK1 QINE FS
☑	CBNK1 QINE MOD
☐	CBNK1 QINE BF

Override Thresholds

		Visible	Identifier	Evaluated Identifier
☒	✓	☑	@defaultLocationId@ QINE @ahpsMinor...	CBNK1 QINE FS
☒	✓	☑	@defaultLocationId@ QINE @ahpsMode...	CBNK1 QINE MOD
☒	✓	☑	@defaultLocationId@ QINE @ahpsMajor...	CBNK1 QINE MAJOR

If the answer to **Question 3** is yes, then proceed to the next **Question**. If no, then the threshold will not be visible. In order for a default threshold to be displayed on a product, it must include an override defined within the **Override Thresholds Table** that has an evaluated identifier that matches it and its visible flag set to “Yes” (next question).

Action: Identify the row of the **Override Thresholds Table** that <u>should</u> correspond to a default threshold to display and select it from the table. For example:



Override Thresholds

		Visible	Identifier	Evaluated Identifier
✕	✓	☑	@defaultLocationId@ QINE @ahpsMinor...	CBNK1 QINE FS
✕	✓	☑	@defaultLocationId@ QINE @ahpsMode...	CBNK1 QINE MOD
✕	✓	☑	@defaultLocationId@ QINE @ahpsMajor...	CBNK1 QINE MAJOR

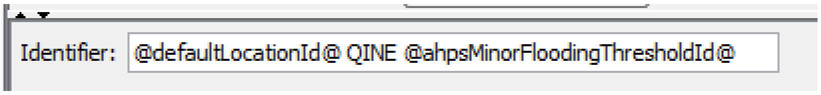
Identifier: @defaultLocationId@ QINE @ahpsMinorFloodingThresholdId@

Visible: Yes

Include in Axis Limits: Default (Yes)

Verify the following:

1. The identifier indicated by the **Identifier Text Field** must be appropriately defined. For example:

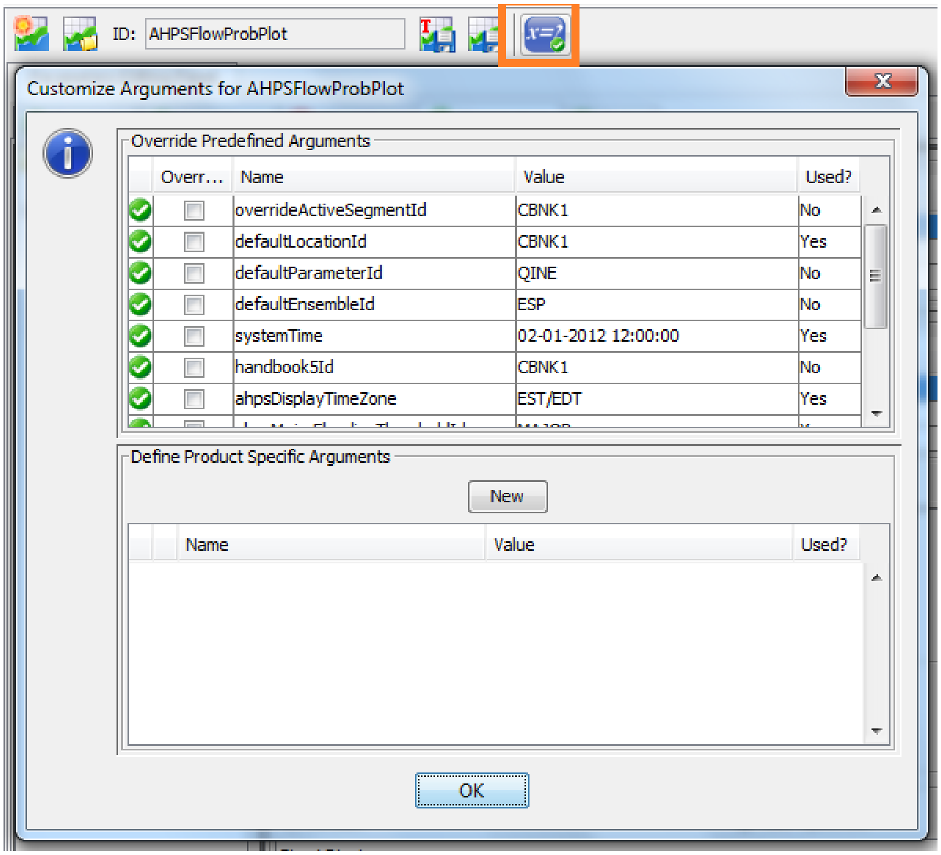
   

   Identifier: @defaultLocationId@ QINE @ahpsMinorFloodingThresholdId@

   NO: Modify the identifier appropriately.

   YES: Go to the next item.

2. If used in the identifier, arguments must be appropriately defined for the active forecast segment. Open the **Customize Global Arguments Dialog** to check the arguments (see the section on the **Customize Global Arguments Dialog** in the *Graphics Generator Reference Manual)*:



ID: AHPSFlowProbPlot

Customize Arguments for AHPSFlowProbPlot

Override Predefined Arguments

	Overr...	Name	Value	Used?
✓	□	overrideActiveSegmentId	CBNK1	No
✓	□	defaultLocationId	CBNK1	Yes
✓	□	defaultParameterId	QINE	No
✓	□	defaultEnsembleId	ESP	No
✓	□	systemTime	02-01-2012 12:00:00	Yes
✓	□	handbook5Id	CBNK1	No
✓	□	ahpsDisplayTimeZone	EST/EDT	Yes

Define Product Specific Arguments

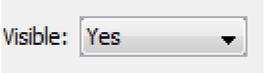
	Name	Value	Used?

NO: Then set the arguments either via the **Customize Global Arguments Dialog**, making the argument product specific, or via the **Modify Settings Dialog** accessible via the button in the **GraphGen Tree Panel** and **GraphGen Thumbnails Panel**. See the section on the **Modify Settings Dialog** in the *Graphics Generator Reference Manual*.

YES: Report this as a bug. If the identifier and arguments are correct, then the override threshold should be paired with the default threshold by Graphics Generator.

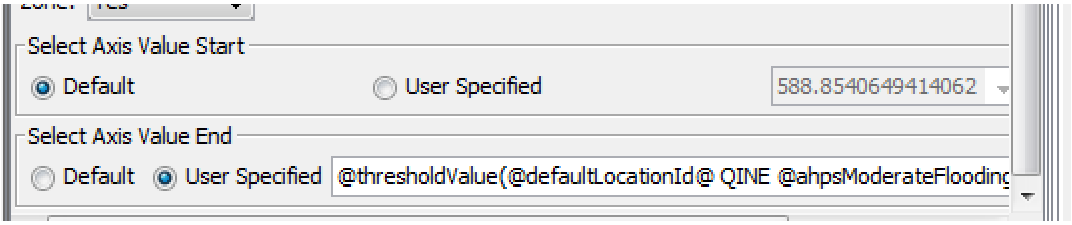
Question 4: Are the threshold parameters set so that the threshold is visible? Select the override threshold from the **Override Thresholds Table** whose evaluated identifier matches the default threshold to display. Verify the following:

1. The **Visible Choicebox** must show that "Yes" is selected:



Visible: Yes

2. The threshold start (and end value for a zone) must be appropriately defined (i.e., not missing and, if arguments are used, able to be evaluated to a number; a log message in the **Logs Panel** will indicate if that is a problem):



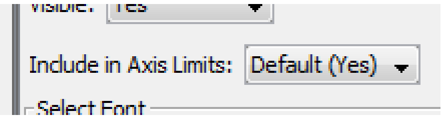
Select Axis Value Start

● Default ○ User Specified 588.8540649414062

Select Axis Value End

○ Default ● User Specified @thresholdValue(@defaultLocationId@ QINE @ahpsModerateFlooding

3. The threshold must either be included in the axis limits,



Include in Axis Limits: Default (Yes)

   or, if not included, the data displayed must be large enough to exceed the threshold allowing it to be viewed. A threshold that is not included in the axis limits will not be visible if the axis range computed based on the data (or via user overrides) does not include the threshold.

A threshold will only be displayed if all three of the above items are true. Modify the parameters as needed (e.g., set the **Visible Choicebox** to be "Yes" and/or the **Include in Axis Limits Choicebox** to be "Yes") to make the threshold visible if you expect it to be visible given the above items.

Question 5: Are the thresholds still not visible? If so, report it as a bug.

3.4.2 A Workaround to Avoid Changing ThresholdValueSets.xml

In order for a threshold to be visible with a time series loaded via the FEWS PI-service, the configuration file

<*configuration_dir*>/RegionConfigFiles/ThresholdValueSets.xml

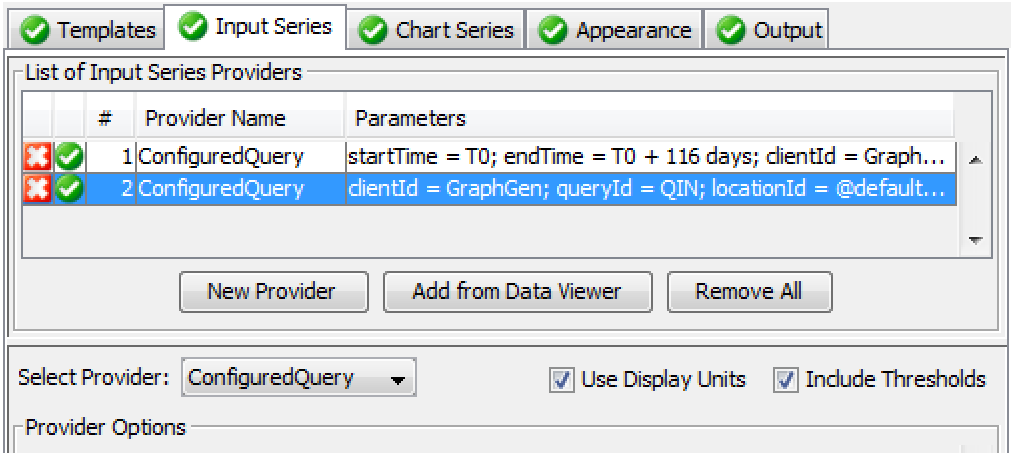
must associate the threshold with the time series. This is done via adding timeSeriesSet XML elements to the configuration file for the appropriate thresholds. However, the ThresholdValueSets.xml file is very particular about the timeSeriesSet XML element: all fields must match. This means that if a threshold must be displayed for an ensemble product, a timeSeriesSet must be added for that ensemble to all locations for which the ensemble product is to be displayed. This may require thousands of lines of additional XML configuration.

There is a straightforward workaround that can be employed *if the thresholds to display are already associated with single-valued time series*; specifically, QIN, QINE, STG, or SSTG time series. The workaround is as follows:

1. Ensure that one of the single-valued time series specified in GraphGen.xml, such as that defined for the QIN, QINE, STG, or SSTG, or time series associated with a query in another file, has associated with it thresholds defined in ThresholdValueSets.xml for the

appropriate forecast segment and data type (flow/stage). Note that the query could yield an all-missing time series; so long as it provides the needed thresholds, it should work. Most RFCs likely already have associated the thresholds with at least one such time series.

2. In the Graphics Generator product of interest, add an input series provider that points to the query identified above via the **Input Series Panel**. The following example is based on a delivered AHPS flow product (see the *Graphics Generator AHPS Product Installation Guide*):



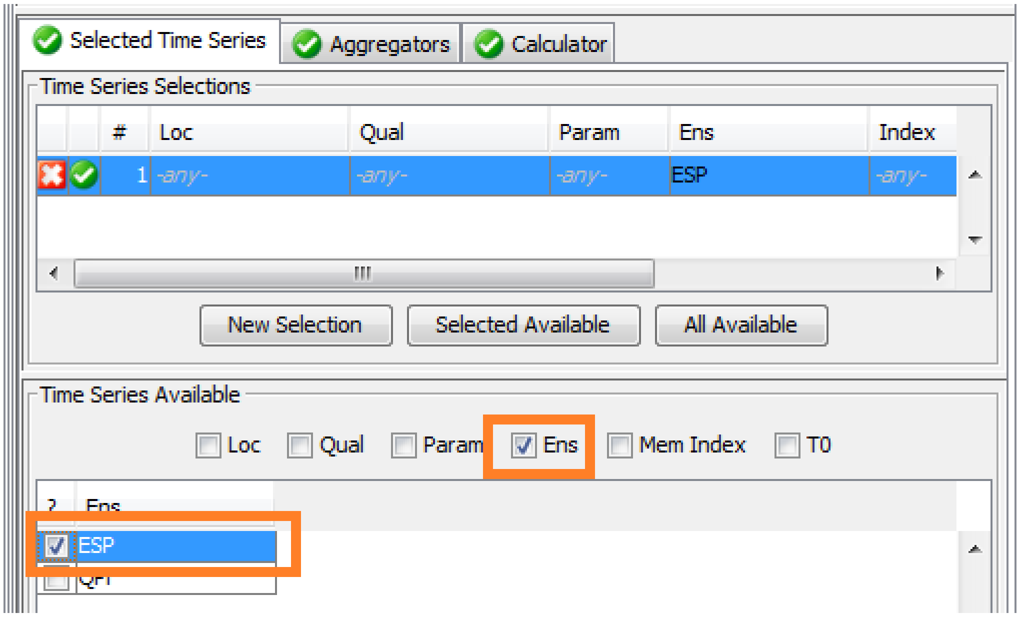
		#	Provider Name	Parameters
☒	✓	1	ConfiguredQuery	startTime = T0; endTime = T0 + 116 days; clientId = Graph...
☒	✓	2	ConfiguredQuery	clientId = GraphGen; queryId = QIN; locationId = @default...

Select Provider: ConfiguredQuery

☑ Use Display Units ☑ Include Thresholds

The **Use Display Units Checkbox** and **Include Thresholds Checkbox** must be checked appropriately so that the data is loaded in the same unit as the other time series to display and the thresholds are included for use in the product.

3. In the **Chart Series Panel**, make sure that the previously defined computations do not include that just added single-valued time series. For example, for the aforementioned AHPS flow product example, this requires defining the time series selection indicated in the **Selected Time Series Subpanel** of the **Chart Series Panel** to required the ensembleId be “ESP”. This can be accomplished by either manually editing the “ENS” column of the **Time Series Selections Table** or by removing the entry in that table (i.e, click on ☒), checking the **Ens Checkbox** above the **Time Series Available Table** and checking the checkbox for the row “ESP”:



#	Loc	Qual	Param	Ens	Index
1	-any-	-any-	-any-	ESP	-any-

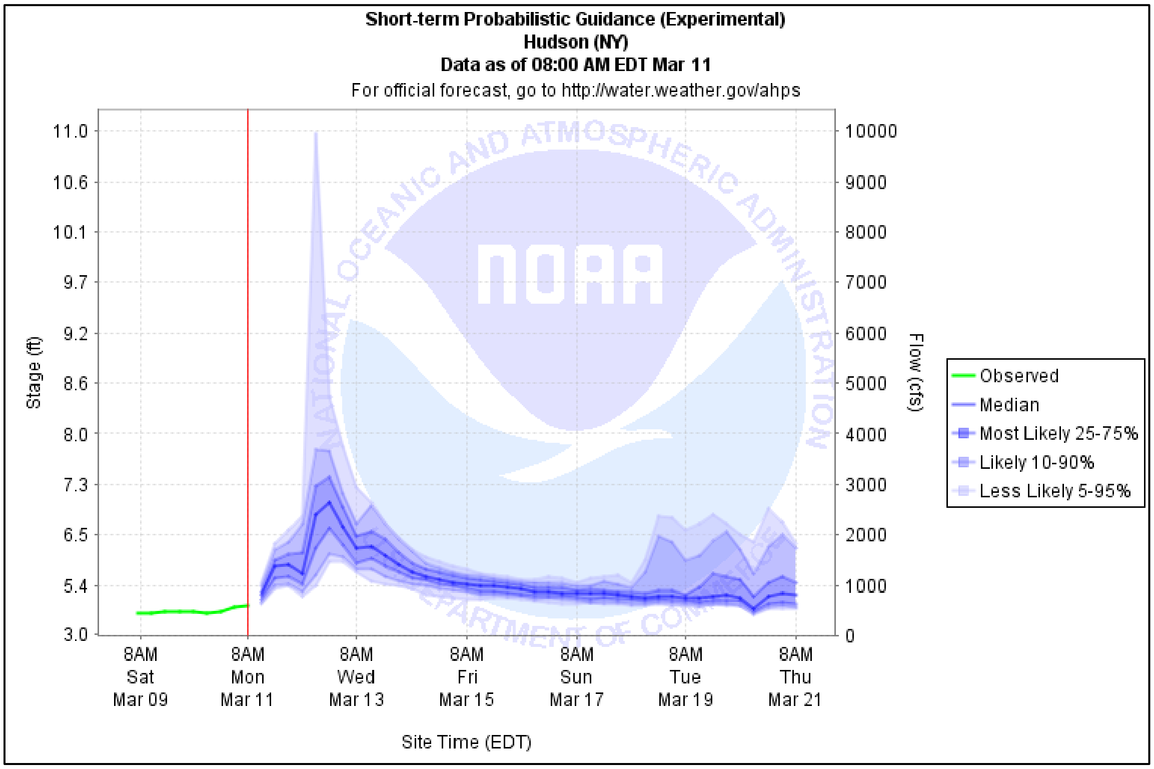
□ Loc □ Qual □ Param ☑ Ens □ Mem Index □ T0

?	Ens
☑	ESP
□	QPF

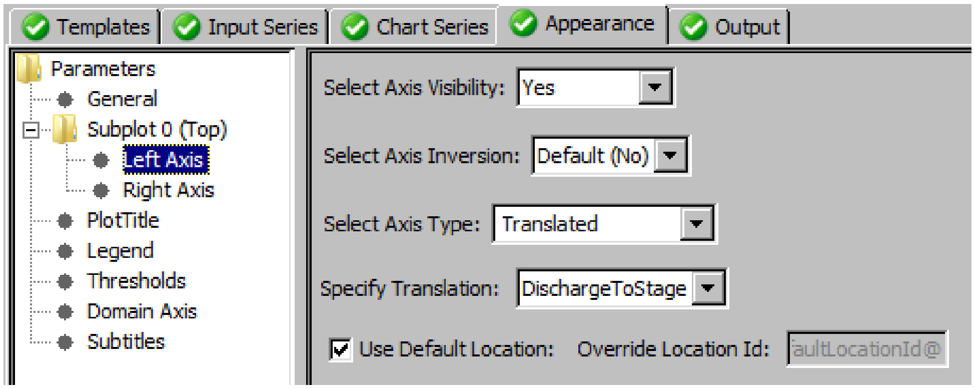
If the appropriate CHPS configuration files are configured correctly, the arguments are defined correctly, and the product is defined correctly to display the threshold (see Section 3.4), then the thresholds should be visible without impacting the computed and displayed chart series.

3.5 *Rating Curve Axis Cannot Be Displayed*

A rating curve axis is a *translated axis* that is displayed opposite of another axis against which data is plotted. For example, in the product,



the displayed data is discharge data and is plotted against the right-axis. The left axis is a translated axis that uses a "DischargeToStage" rating curve translation:



Select Axis Visibility: Yes

Select Axis Inversion: Default (No)

Select Axis Type: Translated

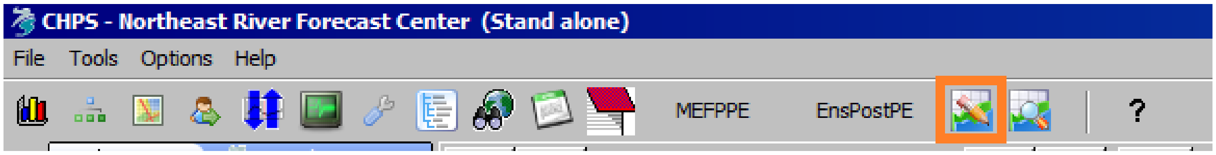
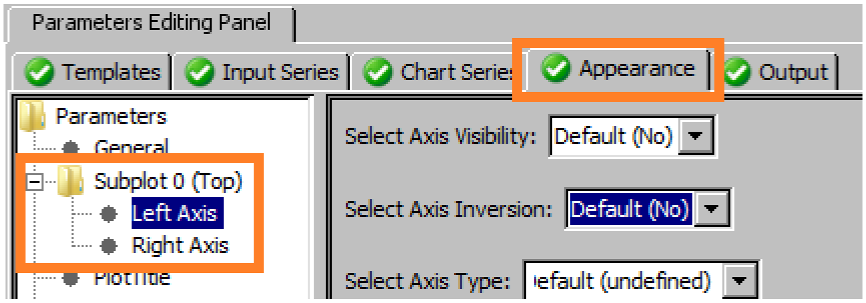
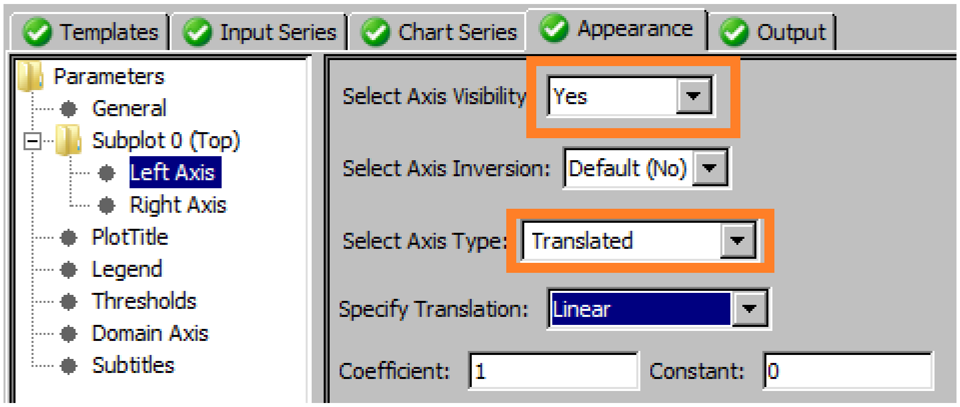
Specify Translation: DischargeToStage

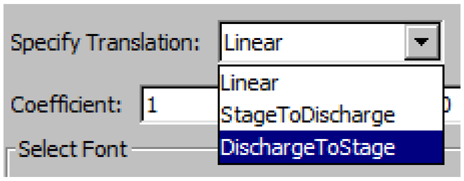
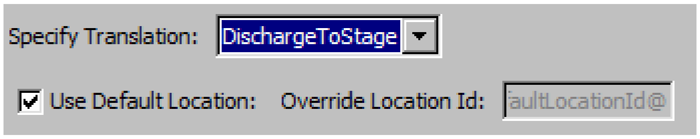
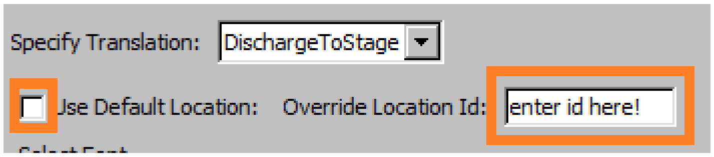
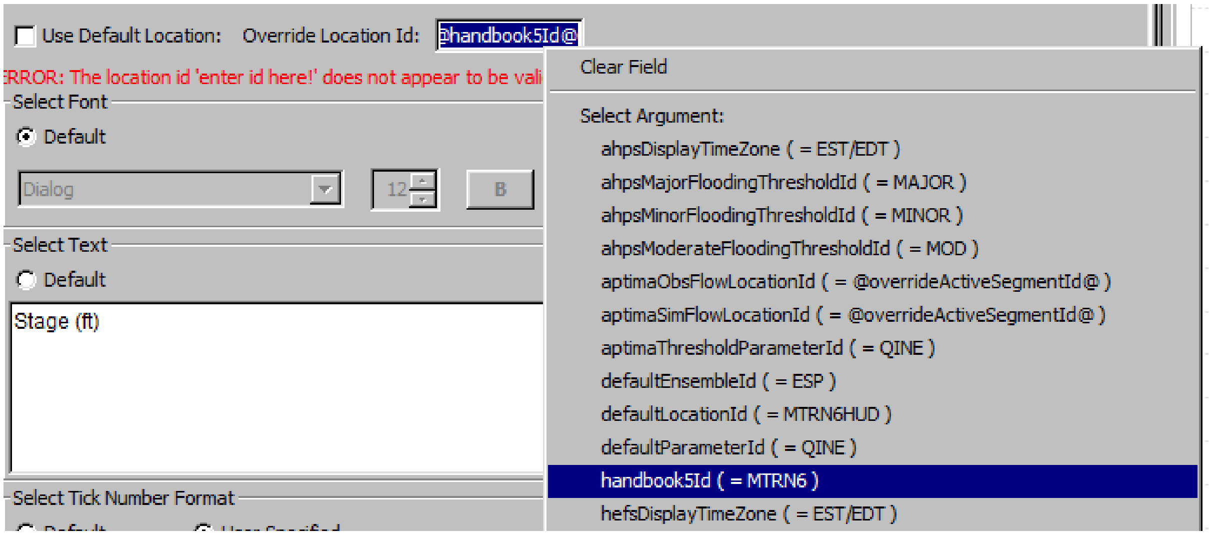
☑ Use Default Location: Override Location Id: aultLocationId@

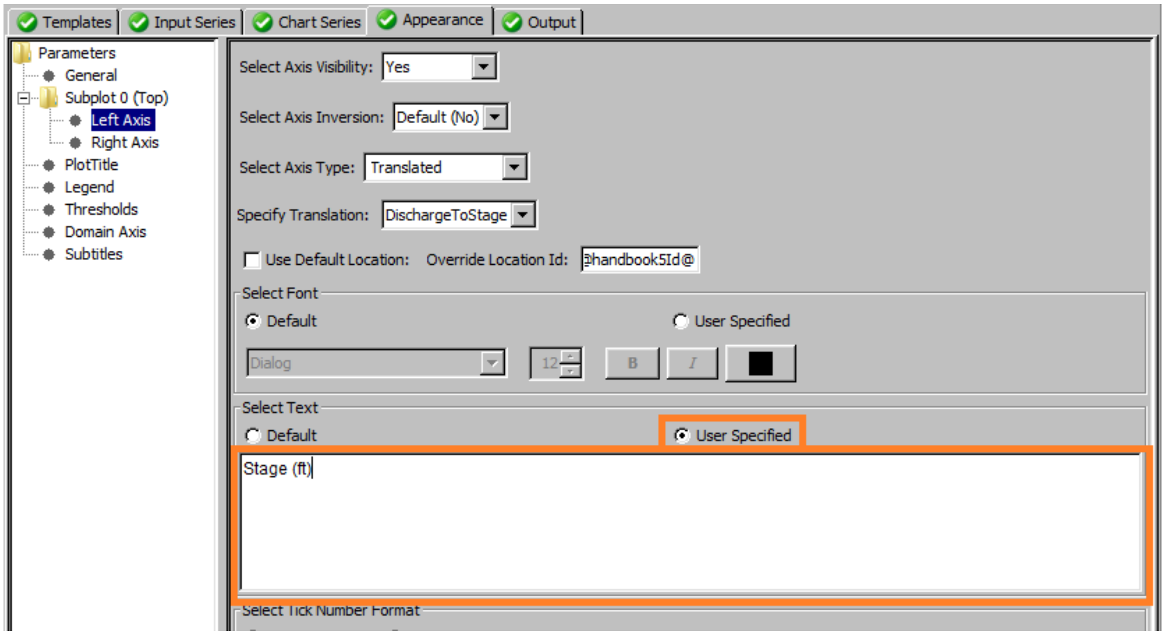
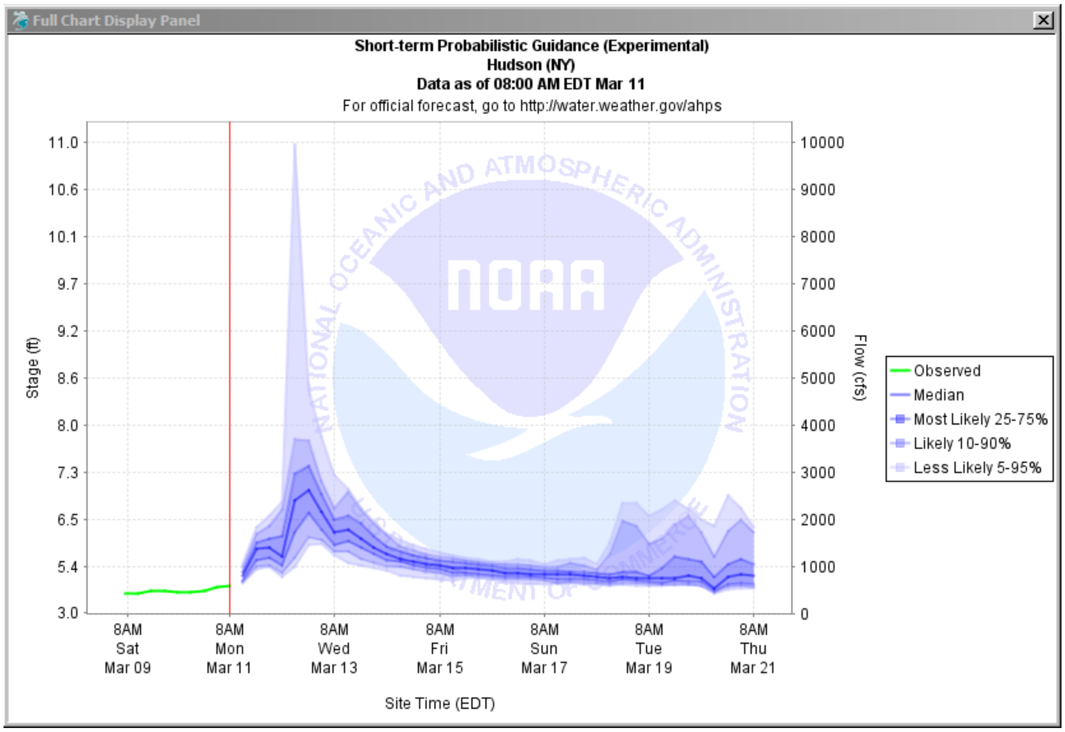
GraphGen applies the rating curve through the FEWS PI-service. Thus, for a rating curve to be accessible to GraphGen, it must (1) be connected to an appropriate PI-service (see Sections 2.3 or the *Graphics Generator Installation Guide* for more information) and (2) that PI-service must have access to the appropriate rating curve.

3.5.1 Adding a Rating Curve Axis

To add the axis to a template, the following steps must be performed:

#	Action
1	Identify the template to modify so that a rating curve axis is displayed.
2	Identify a forecast segment to use while editing the template for which (1) data is displayed and (2) a rating curve is available.
3	Select the forecast segment in the **Forecasts Panel** of the CHPS interface.
4	Open the template for editing in the **GraphGen Editor Panel**. To do so, select the template to modify in the **GraphGen Tree Panel** and open the **GraphGen Editor Panel** by clicking on the **GraphGen Editor Button** in the CHPS toolbar:  CHPS - Northeast River Forecast Center (Stand alone) File Tools Options Help MEFPPE EnsPostPE ?
5	In the **GraphGen Editor Panel**, switch to the **Appearance Tab** and, in the tree on the left, select the axis that is to be the rating curve axis within the subplot that is to display that axis:  Parameters Editing Panel Templates Input Series Chart Serie Appearance Output Parameters General Subplot 0 (Top) Left Axis Right Axis PlotTitle Select Axis Visibility: Default (No) Select Axis Inversion: Default (No) Select Axis Type: efault (undefined)
6	Set the axis visibility to “Yes” and, in the **Select Axis Type Choicebox**, select “Translated”:  Templates Input Series Chart Series Appearance Output Parameters General Subplot 0 (Top) Left Axis Right Axis PlotTitle Legend Thresholds Domain Axis Subtitles Select Axis Visibility Yes Select Axis Inversion: Default (No) Select Axis Type: Translated Specify Translation: Linear Coefficient: 1 Constant: 0 In the screenshot above, the components below the **Select Axis Type Choicebox** willappear once the axis type is selected.

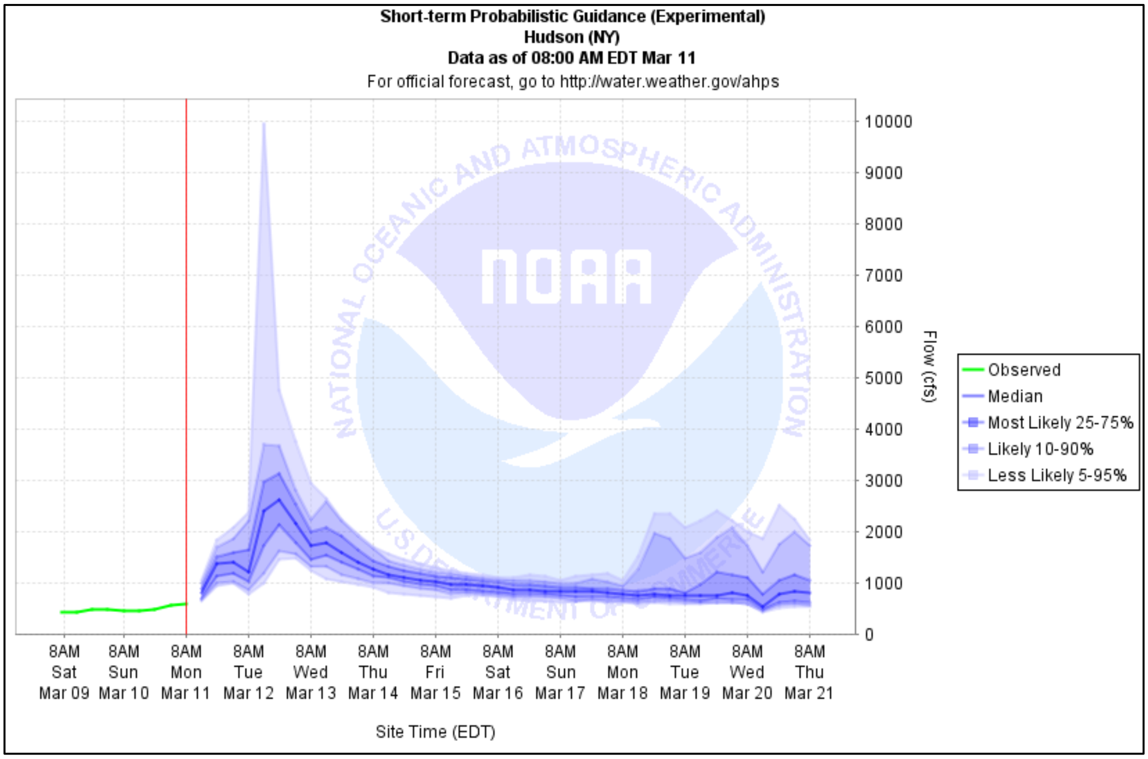
#	Action
7	From the **Specify Translation Choicebox**, select the appropriate translation:  Specify Translation: Linear Coefficient: 1 Select Font Linear StageToDischarge DischargeToStage In the example above, a DischargeToStage translation is selected, implying that the opposite axis (against which data is plotted) displays discharge (flow) data. Use the “StageToDischarge” translation if the opposite axis displays stage data. After selection, components will be displayed below the **Translation Choicebox** allowing for specifying the locationId to use to acquire the rating curve:  Specify Translation: DischargeToStage ☑ Use Default Location: Override Location Id: aultLocationId@
8	If the argument defaultLocationId, which typically corresponds to the segment id, is not correct for acquiring the rating curve, modify the **Override Location Id Field** by, first, unchecking the **Use Default Location Checkbox** and then editing the field appropriately:  Specify Translation: DischargeToStage ☐ Use Default Location: Override Location Id: enter id here! In particular, if rating curves are stored in the database with 5-character ids whereas the defaultLocationId may be 8 characters, then set the **Override Location Id Field** to use the handbook5Id argument. This can be done by right-clicking on the field, clicking on “handbook5Id” in the drop-down menu, and pressing <Enter>:  ☐ Use Default Location: Override Location Id: @handbook5Id@ RROR: The location id 'enter id here!' does not appear to be vali Select Font ● Default Dialog 12 B Select Text ○ Default Stage (ft) Select Tick Number Format Clear Field Select Argument: ahpsDisplayTimeZone (= EST/EDT) ahpsMajorFloodingThresholdId (= MAJOR) ahpsMinorFloodingThresholdId (= MINOR) ahpsModerateFloodingThresholdId (= MOD) aptimaObsFlowLocationId (= @overrideActiveSegmentId@) aptimaSimFlowLocationId (= @overrideActiveSegmentId@) aptimaThresholdParameterId (= QINE) defaultEnsembleId (= ESP) defaultLocationId (= MTRN6HUD) defaultParameterId (= QINE) handbook5Id (= MTRN6) hefsDisplayTimeZone (= EST/EDT)

#	Action
9	Modify the axis title appropriately via the **Select Text Panel** below lower in the parameter editing panel. Click on the **User Specified Radio Button** and modify the text appropriately:  Templates | Input Series | Chart Series | Appearance | Output Parameters General Subplot 0 (Top) Left Axis Right Axis PlotTitle Legend Thresholds Domain Axis Subtitles Select Axis Visibility: Yes Select Axis Inversion: Default (No) Select Axis Type: Translated Specify Translation: DischargeToStage ☐ Use Default Location: Override Location Id: @handbook5Id@ Select Font ● Default ○ User Specified Dialog 12 B I Select Text ○ Default ● User Specified Stage (ft) Select Tick Number Format
10	Confirm that the rating curve axis is displayed in the **Full Chart Display Panel** as desired: 
11	Save the template appropriately. Use the **GraphGen Viewer Panel** to view the product and confirm that the rating curve axis is appropriately displayed. Upload the templates to the central area by clicking on the **Upload Button** in the **GraphGen Tree Panel**.

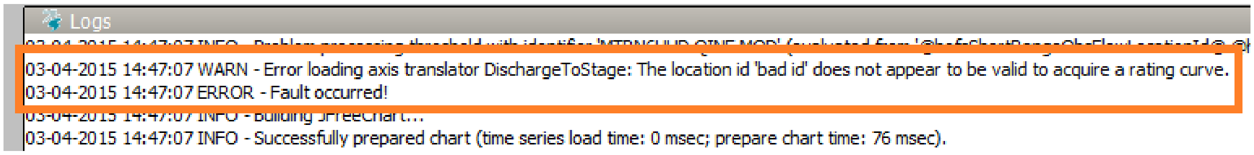
3.5.2 Rating Curve Axis is Not Appropriately Displayed

If a template is defined to display a rating curve, but it cannot be displayed for a particular active forecast segment, then the following will occur:

1. The displayed product chart will not include the corresponding axis. This is equivalent to if the axis visibility is set to false. For example,

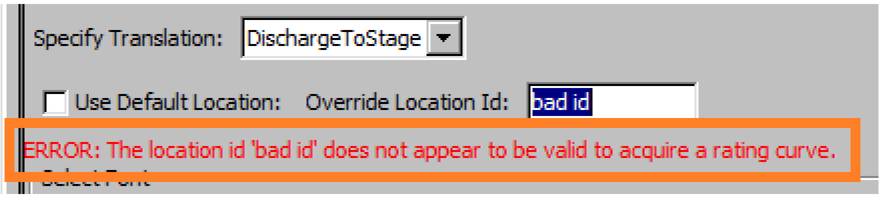


2. A warning message (along with an "ERROR – Fault occurred!" message generated by the FEWS PI-service) will be displayed in the CHPS **Logs Panel** or in the diagnostics log output of the GraphGenModelAdapter:



```
03-04-2015 14:47:07 INFO - Problem processing threshold with identifier 'MTRNCHUD_QINE_MOD' (evaluated from '@hefsShortRangeObsFlowLocationId@_@I
03-04-2015 14:47:07 WARN - Error loading axis translator DischargeToStage: The location id 'bad id' does not appear to be valid to acquire a rating curve.
03-04-2015 14:47:07 ERROR - Fault occurred!
03-04-2015 14:47:07 INFO - Building JFreeChart...
03-04-2015 14:47:07 INFO - Successfully prepared chart (time series load time: 0 msec; prepare chart time: 76 msec).
```

3. If viewed in the **GraphGen Editor Panel**, an error message will be displayed under the **Translation Choicebox** and locationId identifying components:



Specify Translation: DischargeToStage

□ Use Default Location: Override Location Id: bad id

ERROR: The location id 'bad id' does not appear to be valid to acquire a rating curve.

The following are the likely causes and associated solutions if the rating curve axis cannot be displayed on the product:

1. **PROBLEM:** The GraphGen components are not connected to the appropriate FEWS PI-service.
   SOLUTION: Identify the appropriate FEWS PI-service either (1) via setting the port number if viewed interactive (see Section 2.3), or (2) setting run file properties appropriate if generated via execution of the GraphGenModelAdapter (see the *Graphics Generator Installation Guide* for more information).

2. **PROBLEM:** The FEWS PI-service does not include an appropriate rating curve.
   SOLUTION: Either import or define the appropriate rating curve; ask your CHPS configuration focal point for more information.

3. **PROBLEM:** The locationId used to acquire the rating curve is not correct.
   SOLUTION: Modify the **Override Location Id Field** to be appropriately specified. See Step 8 in Section 3.5.1 above.

3.6 *Location Information (Description, Latitude, etc.) Is Not Displayed*

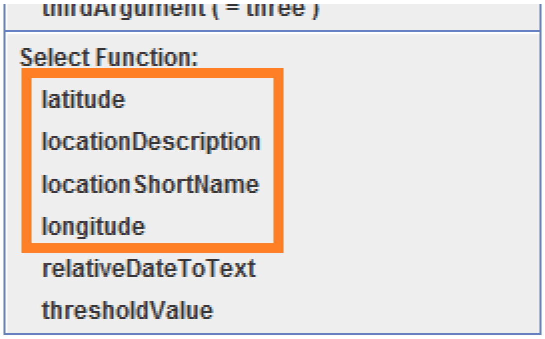
Graphics Generator acquires the location information via the FEWS PI-service. In order to acquire the information, a filter defined within the file

<configuration_dir>/RegionConfigFiles/Filters.xml

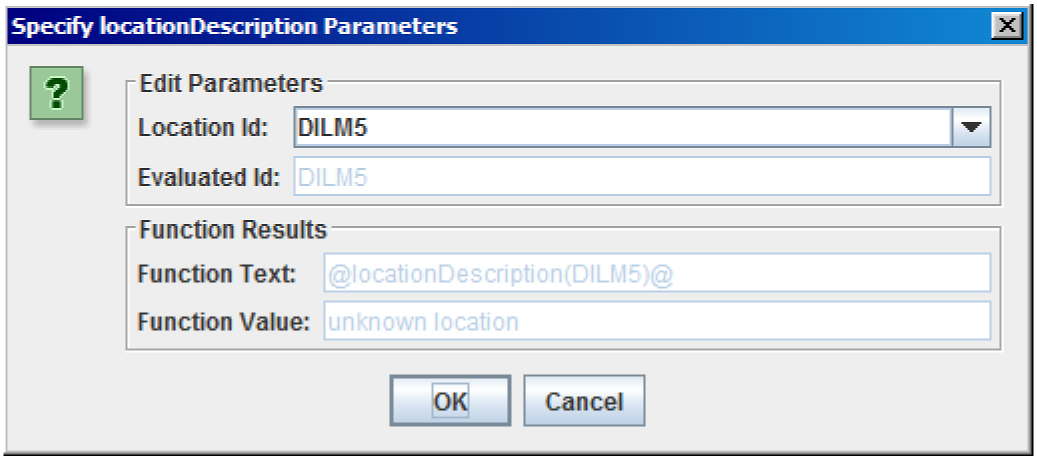
must be provided. For Graphics Generator, that filter is defined with the id XML element of "GraphGen" and is setup as part of the installation process described in the *Graphics Generator Install Guide*.

GraphGen will only have information to location information for any locationId referred to within a timeSeriesSet defined in the filter with id "GraphGen". For that reason, the "GraphGen" filter should include all time series for which products may ever need to be generated.

The location information is added to templates by way of *argument functions* which can be accessed by right-clicking on components within the **GraphGen Editor Panel** that allow for arguments. A right-click will result in a popup menu being displayed including the following menu items:



Clicking on any of these four menu items will result in a dialog opening similar to the following:


Specify locationDescription Parameters

Edit Parameters

Location Id: DILM5

Evaluated Id: DILM5

Function Results

Function Text: @locationDescription(DILM5)@

Function Value: unknown location

The locationId for which to acquire the location information must be specified in the **Location Id Field** at the top. The remaining components display the **Evaluated Id** (if an argument is used to define the locationId) and the resulting argument **Function Text** and replacement **Function Value**. The argument function value will be the value of the corresponding XML element in the configuration file

<*configuration_dir*>/RegionConfigFiles/Locations.xml

defined for the evaluated id. For example, @locationDescription(CBNK1)@ will return the value of the description XML element specified in the Locations.xml file for the location XML element with id attribute set to "CBNK1":

```
<locations …>
…
   <location id="CBNK1" name="CBNK1">
      <description>CBNK1 - CHIKASKIA RIVER AT CORBIN 3W</description>
      <shortName>CBNK1</shortName>
      <toolTip>… removed …</toolTip>
      <x>-97.6016666667</x>
      <y>37.1291666667</y>
      <z>342.9</z>
   </location>
…
</locations>
```

In this case, the returned value will be "CBNK1 - CHIKASKIA RIVER AT CORBIN 3W".

If location information cannot be displayed for a given selected active forecast segment and template, the following are the likely causes (with associated solutions):

1. **PROBLEM:** The locationId, after evaluation, being passed to the argument function is incorrect.
   SOLUTION: Correct that locationId by modifying the template parameters appropriately.

2. **PROBLEM:** No filter with id "GraphGen" is included in the Filters.xml.
   SOLUTION: Add the "GraphGen" filter to the Filters.xml file; see the *Graphics Generator Installation Guide*.

3. **PROBLEM:** The "GraphGen" filter defined in Filters.xml does not include the evaluated locationId being passed to the argument function.
   SOLUTION: Modify the "GraphGen" filter appropriately so that an included timeSeriesSet has a locationId identical to the desired locationId.